```
                                title   'DDT86 1.1  10/2/81'
                        ;
                        ;       modified  5/14/81  R. Silberstein
                        ;       modified  6/15/81  R. Silberstein
                        ;       modified  8/12/81  R. Silberstein
                        ;       modified  9/6/81   R. Silberstein
                        ;       modified  9/16/81  R. Silberstein
                        ;       modified  10/1/81  R. Silberstein
                        ;
                        ;
                        ;       ****************************************
                        ;       *                                      *
                        ;       *      D D T  8 0 8 6 - 8 0 8 8        *
                        ;       *                                      *
                        ;       ****************************************
                        ;
0000                    debug           equ     00h     ;if set, use direct bios calls for console io
                        ;
0100                    ddt_org         equ     100h            ;origin of this module
1400                    lasmorg         equ     ddt_org+1300h   ;origin of disassembler
2300                    asmorg          equ     ddt_org+2200h   ;origin of assembler
                        ;
                                        cseg
                        ;
                                        org     005ch
005C                    fcb             rb      10h
006C                    fcb2            rb      14h
0080                    buff            rb      30h
                        ;
                                        org     lasmorg
1400 B80000             disem:          mov     ax,0
1403 C20400                             ret     4       ;remove parameters from stack
                        ;
                                        org     asmorg
2300 B80000             assem:          mov     ax,0
2303 C20400                             ret     4       ;remove parameters from stack
                        ;
                                        org     ddt_org
0100 E94A01     024D                    jmp     ddt86   ;ccp transfers control here
0103 E9EB02     03F1                    jmp     conin
0106 E9BB02     03C4                    jmp     plmconout
0109 E9C502     03D1                    jmp     plmgetline
010C E94B0A     0B5A                    jmp     asment  ;get here on error in assem (pl/m)
010F E94508     0957                    jmp     plmset  ;assembler link to ddt set/verify
                        ;
0112 CDE0               bdosint:        int     bdosi   ;this is only here for user to patch
                                                        ;actual bdos link gets set from here
0112                    bdosintloc      equ     offset bdosint
0113                    bdosintnum      equ     offset bdosint+1
                        ;
00E0                    bdosi           equ     224     ;bdos interrupt number
```

CP/M-86 v.1.1 (Vol. III)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
                                
0060                stsize      equ     96      ;stack size
0002                nbps        equ     2       ;number of breakpoints
                    ;
0200                ifmask16    equ     0200h   ;16-bit IF mask
0002                ifmask8     equ     02h     ;8-bit IF mask
                    ;
000A                lf          equ     0ah     ;line feed
000D                cr          equ     0dh     ;carriage return
000D                eol         equ     cr
0013                ctls        equ     13h     ;ascii ctl-s
                    ;
                    ;   ******************************************
                    ;   *                                        *
                    ;   *          m e s s a g e s               *
                    ;   *                                        *
                    ;   ******************************************
                    ;
0114 20434F505952   copyright   db      ' COPYRIGHT (C) 1981, DIGITAL RESEARCH '
     494748542028
     432920313938
     312C20444947
     4954414C2052
     455345415243
     4820
                    ;
013A 444454383620   signon      db      'DDT86 1.','1' or 80h
     312EB1
                    ;
0143 2031302F3032   DATE        DB      ' 10/02/81 '
     2F383120
014B 41D8           regname     db      'A','X' or 80h
014F 42D8                       db      'B','X' or 80h
0151 43D8                       db      'C','X' or 80h
0153 44D8                       db      'D','X' or 80h
0155 53D0                       db      'S','P' or 80h
0157 42D0                       db      'B','P' or 80h
0159 53C9                       db      'S','I' or 80h
015B 44C9                       db      'D','I' or 80h
015D 43D3           segreg      db      'C','S' or 80h
015F 44D3                       db      'D','S' or 80h
0161 53D3                       db      'S','S' or 80h
0163 45D3                       db      'E','S' or 80h
0165 49D0                       db      'I','P' or 80h
                    ;
0167 4F4449545A5A   flagname    db      'ODITSZAPC'
     415043
0170 05060708090A   flagbits    db      5,6,7,8,9,10,12,14,16
     0C0E10
                    ;
0179 43D3           segnames    db      'C','S' or 80h
017B 44D3                       db      'D','S' or 80h
017D 45D3                       db      'E','S' or 80h
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                ;
017F 53D3                       db      'S','S' or 80h
0181 53B1                       db      'X','1' or 80h
0183 58B2                       db      'X','2' or 80h
0185 58B3                       db      'X','3' or 80h
0187 58B4                       db      'X','4' or 80h
                        ;
0189 0D0A43414E4E       closem  db      cr,lf,'CANNOT CLOS','E' or 80h
     4F5420434C4F
     53C5
0197 0D0A494E5355       loadm   db      cr,lf,'INSUFFICIENT MEMOR','Y' or 80h
     464649434945
     4E54204D454D
     4F52D9
01AC 0D0A4E4F2053       makem   db      cr,lf,'NO SPAC','E' or 80h
     504143C5
01B6 0D0A4D454D4F       memm    db      cr,lf,'MEMORY REQUEST DENIE','D' or 80h
     525920524551
     554553542044
     454E4945C4
01CB 0D0A4E4F2046       openm   db      cr,lf,'NO FIL','E' or 80h
     494CC5
01D6 202053544152       readm   db      '  START      EN','D' or 80h
     542020202020
     20454EC4
01E6 0D0A56455249       verm    db      cr,lf,'VERIFY ERROR AT',' ' or 80h
     465920455252
     4F5220415420
01F8 0D0A4449534B       writem  db      cr,lf,'DISK WRITE ERRO','R' or 80h
     205752495445
     204552524FD2
                        ;
                        ;       ************************************************
                        ;       *                                              *
                        ;       *         i n i t i a l i z a t i o n          *
                        ;       *                                              *
                        ;       ************************************************
                        ;
                        setbdosint:                     ;copy vector at 0:bdosi*4 to (bdosi+1)*4
020A A01301                     mov     al,byte ptr .bdosintnum ;get bdos interrupt #
020D FEC0                       inc     al
020F A21E03                     mov     byte ptr .ddtbdosintnum,al      ;ddt uses the next interrupt internal
0212 2AE4                       sub     ah,ah
0214 D1E0                       shl     ax,1
0216 D1E0                       shl     ax,1            ;bdos int # * 4
0218 8BF8                       mov     di,ax           ;[di] points to new bdos interrupt vector
021A 8BF0                       mov     si,ax
021C 83EE04                     sub     si,4            ;[si] points to actual bdos interrupt vector
021F 1E                         push    ds              ;save ds
0220 2BC0                       sub     ax,ax
0222 8EC0                       mov     es,ax           ;set es and ds to 0 to move interrupt vectors
0224 8ED8                       mov     ds,ax
0226 B90400                     mov     cx,4
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

```
 160
 161  0229 F3A4              rep     movs    al,al           ;copy bdos interrupt vector to next int vector
 162  022B 1F                        pop     ds              ;restore ds
 163  022C C3                        ret
 164                         ;
 165                         checkcmdtail:                   ;if command tail not empty, assume E command
 166  022D BE8000                    mov     si,offset buff
 167  0230 B400                      mov     ah,0
 168  0232 AC                        lods    al              ;get count from command tail
 169  0233 0AC0                      or      al,al
 170  0235 7415              024C    jz      cctret          ;nothing to do if tail empty
 171  0237 3C40                      cmp     al,conbuffmax
 172  0239 7602              023D    jbe     movcom
 173  023B B040                      mov     al,conbuffmax   ;truncate, if needed
 174                             movcom:
 175  023D 8BC8                      mov     cx,ax           ;count to [cx]
 176  023F BFC311                    mov     di,offset conbuff
 177  0242 1E                        push    ds
 178  0243 07                        pop     es              ;point destination to ddt seg
 179  0244 F3A4              rep     movs    al,al           ;copy command tail into ddt command buff
 180  0246 B00D                      mov     al,eol
 181  0248 AA                        stos    al              ;store terminator
 182  0249 E84B0A            0C97    call    execute         ;command tail is assumed E command
 183                             cctret:
 184  024C C3                        ret
 185                         ;
 186                         ddt86:
 187  024D FC                        cld
 188  024E 3C16BD11                  MOV     CCPSS,SS
 189  0252 8926BF11                  MOV     CCPSP,SP        ;SAVE CCP STACK POINTER
 190  0256 2E8C16B011                MOV     USERSS,SS
 191  025B 2E8926A411                MOV     USERSP,SP       ;INITIALIZE USER'S MACHINE STATE TO CCP STACK
 192                         ;
 193  0260 9C                        pushf
 194  0261 58                        pop     ax              ;get flags
 195  0262 250002                    and     ax,ifmask16     ;mask to IF bit
 196  0265 88266B12                  mov     sysif,ah        ;save system IF state
 197  0269 2EA3B611                  mov     userfl,ax       ;initialize user's flags to sysif
 198                         ;
 199  026D 8CC8                      mov     ax,cs
 200  026F FA                        cli                     ;entering critical region
 201  0270 8ED0                      mov     ss,ax           ;set ss = cs
 202  0272 BC1013                    mov     sp,offset stackp        ;set up stack pointer
 203  0275 F6066B1202                test    sysif,ifmask8   ;see if interrupts were enabled
 204  027A 7401              027D    jz      d0              ;don't turn them on if they were off
 205  027C FB                        sti
 206                             d0:                         ;exiting critical region
 207                         ;
 208  027D E88AFF            020A    call    setbdosint      ;copy vector since ddt uses bdosif1 internally
 209                         ;
 210  0280 F6066F12FF                test    savevecflag,0ffh        ;ddt interrupts saved on each g/t/u?
 211  0285 7503              028A    jnz     d1                      ;if so, don't initialize here
 212  0287 E8DA04            0764    call    bpvect                  ;if not, initialize them here
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
S/N: #

```
                        d1:
                                if      debug
                        ;
                                sub     ax,ax
                                mov     es,ax
                                mov     si,bdosi * 4 + 2
                                mov     ax,es:[si]      ;get bdos segment
                                mov     es,ax
                                mov     biosentryseg,ax
                                mov     di,biosentryoff
                                mov     al,81h
                                stos    al
                                mov     al,0c3h
                                stos    al
                                mov     al,0h
                                stos    al
                                mov     al,25h
                                stos    al
                                mov     al,0ffh
                                stos    al
                                mov     al,0d3h
                                stos    al
                                mov     al,0cbh
                                stos    al
                        ;
                                endif
                        ;
028A BE3A01                     mov     si,offset signon        ;get sign on message
028D E8AD01     043D            call    printm          ;and print it
                        ;
0290 E8A700     033A            CALL    VERSION
0293 3C30                       CMP     AL,30H          ;SEE IF WE ARE UNDER FILE SYSTEM III (MP/M)
0295 B700                       MOV     BH,0
0297 7207       02A0            JC      D2              ;IF EARLIER VERSION, SKIP
0299 B2FE                       MOV     DL,0FEH
029B E8EC00     038A            CALL    SETERRMODE      ;SO BDOS RETURNS TO DDT ON FILE ERRORS
029E B701                       MOV     BH,1
                        D2:
02A0 883E7112                   MOV     ERRMODE,BH
                        ;
02A4 E886FF     022D            call    checkcmdtail    ;if non-blank, do E command
                        ;
                        ;       ************************************************
                        ;       *                                              *
                        ;       *         w o r k i n g   l o o p              *
                        ;       *                                              *
                        ;       ************************************************
                        ;
02A7 FC                 start:  cld                     ;direction flag points up
02A8 BC1013                     mov     sp,offset stackp        ;make sure stack is right
02AB E86F01     041D            call    crlf            ;print crlf
02AE B02D                       mov     al,'-'          ;and prompt
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                                                    ;get command line
02B0 E87300      0326          call    conout
02B3 E82101      03D7          call    getline         ;get command line
02B6 E83801      03F1          call    conin           ;read first char
02B9 3C0D                      cmp     al,eol
02BB 74EA        02A7          jz      start
02BD 3C3B                      CMP     AL,';'
02BF 74E6        02A7          JZ      START           ;IGNORE COMMENT LINES
02C1 2C41                      sub     al,'A'          ;check range for valid command
02C3 721A        02DF          jb      err
02C5 3C19                      cmp     al,'Z'-'A'
02C7 7716        02DF          ja      err
02C9 D0E0                      shl     al,1            ;* 2 (2 bytes per ctable entry)
02CB B400                      mov     ah,0
02CD 93                        xchg    ax,bx
02CE C6066C1201                mov     numreq,1        ;most commands require an argument
02D3 C6066D1200                mov     wmode,0         ;most commands are not word mode
02D8 2EFF97E902                call    word ptr ctable[bx]     ;immed call command routine
02DD EBC8        02A7          jmps    start           ;start over
                           ;
                           err:
02DF E83B01      041D          call    crlf
02E2 B03F                      mov     al,'?'          ;error handler
02E4 E83F00      0326          call    conout          ;print error char
02E7 EB8E        02A7          jmps    start           ;stack maybe messed up, keep this jmp
                           ;
                           ;   ************************************************
                           ;   *                                              *
                           ;   *      c o m m a n d   j u m p   t a b l e     *
                           ;   *                                              *
                           ;   ************************************************
                           ;
02E9 2F0B                  ctable  dw      assm            ;assemble mnemonics
02EB 300B                          DW      BLOCKCOMPARE    ;COMPARE MEMORY BLOCKS
02ED DF02                          dw      err
02EF D60B                          dw      display         ;display memory
02F1 970C                          dw      execute         ;load user program for execution
02F3 150D                          dw      fill            ;fill memory with constant
02F5 5B0D                          dw      gouser          ;go to user program
02F7 BF0D                          dw      hexmath         ;compute hex sum and difference
02F9 DE0D                          dw      ifcb            ;input file control block
02FB DF02                          dw      err
02FD DF02                          dw      err
02FF 2E0E                          dw      lassm           ;disassemble memory
0301 A40E                          dw      move            ;move block
0303 DF02                          dw      err
0305 DF02                          dw      err
0307 DF02                          dw      err
0309 DF02                          dw      err
030B F20E                          dw      read            ;read file
030D 4C0F                          dw      setmem          ;set memory
030F A10F                          dw      trace           ;trace program execution
0311 D70F                          dw      untrace         ;untraced program execution
```

COPYRIGHT © 1981 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                        dw      verify          ;display file info
0313 DE0F               dw      write           ;write memory block to disk
0315 3010               dw      xcom            ;display/alter CPU state
0317 E410               dw      err
0319 DF02               dw      err
031B DF02               dw      err
                        ;
                        ;       ************************************************
                        ;       *                                              *
                        ;       *        b d o s   i n t e r f a c e           *
                        ;       *                                              *
                        ;       ************************************************
                        ;
                        bdos:                           ;this interrupt instruction is overwritten on initial
                                                        ;the actual int # used is the one at bdosint: + 1
  031E                  ddtbdosintnum   equ     offset bdos + 1
                        ;
031D CDE0               int     bdosi
031F C3                 ret
                        ;
                        if      debug
                        ;
                        bios:
                        callf   dword ptr biosentryoff
                        ret
                        ;
                        endif
                        ;
                        if      debug
                        ;
                        consin:
                        mov     bx,9
                        call    bios
                        push    ax
                        call    conout
                        pop     ax
                        ret
                        ;
                        endif
                        ;
                        if      not debug
                        ;
                        consin:
0320 B101               mov     cl,1
0322 E8F8FF     031D    call    bdos
0325 C3                 ret
                        ;
                        endif
                        ;
                        if      debug
                        ;
                        conout:
                        mov     bx,0ch
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                        mov     cl,al
                                        jmp     bios
                                ;
                                        endif
                                ;
                                        if      not debug
                                ;
                                conout:
0326 B102                               mov     cl,2
0328 8AD0                               mov     dl,al
032A E8F0FF             031D            call    bdos
032D C3                                 ret
                                ;
                                        endif
                                ;
                                rdconbuff:
032L B10A                               mov     cl,10
0330 E8EAFF             031D            call    bdos
0333 C3                                 ret
                                ;
                                        if      debug
                                ;
                                constat:
                                        mov     bx,6
                                        jmp     bios
                                ;
                                        endif
                                ;
                                        if      not debug
                                ;
                                constat:
0334 B10B                               mov     cl,11
0336 E8E4FF             031D            call    bdos
0339 C3                                 ret
                                ;
                                        endif
                                ;
                                VERSION:
033A B10C                               MOV     CL,12
033C E9DEFF             031D            JMP     BDOS
                                ;
                                open:
033F B10F                               mov     cl,15
0341 E8D9FF             031D            call    bdos
0344 FEC0                               inc     al              ;test for 0ffh returned
0346 7401               0349            jz      openerr
0348 C3                                 ret
                                openerr:
0349 BECD01                             mov     si,offset openm
034C EB70               03BE            jmps    errm
                                ;
                                close:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                mov     cl,16
034E B110                       call    bdos
0350 E8CAFF     031D            inc     al
0353 FEC0                       jz      closeerr
0355 7401       0358            ret
0357 C3                         closeerr:
                                mov     si,offset closem
0358 BE8901                     jmps    errm
035B EB61       03BE            ;
                                delete:
                                mov     cl,19
035D B113                       jmp     bdos
035F E9B8FF     031D            ;
                                readsec:
                                mov     cl,20
0362 B114                       jmp     bdos
0364 E9B6FF     031D            ;
                                writesec:
                                mov     cl,21
0367 B115                       call    bdos
0369 E8B1FF     031D            or      al,al
036C 0AC0                       jnz     writeerr
036E 7501       0371            ret
0370 C3                         writeerr:
                                mov     si,offset writem
0371 BEF801                     jmps    errm
0374 EB48       03BE            ;
                                make:
                                mov     cl,22
0376 B116                       call    bdos
0378 E8A2FF     031D            inc     al
037B FEC0                       jz      makeerr
037D 7401       0380            ret
037F C3                         makeerr:
                                mov     si,offset makem
0380 BEAC01                     jmps    errm
0383 EB39       03BE            ;
                                setdma:
                                mov     cl,26
0385 B11A                       jmp     bdos
0387 E993FF     031D            ;
                                SETERRMODE:
                                MOV     CL,45
038A B12D                       JMP     BDOS
038C E98EFF     031D            ;
                                setdmab:
                                mov     cl,51
038F B133                       jmp     bdos
0391 E989FF     031D            ;
                                getmaxmem:
                                mov     cl,53
0394 B135                       call    bdos
0396 E884FF     031D
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                inc     al
0399 FEC0                       jz      memerr
039B 740B       03A8            ret
039D C3                         ;
                                allocabsmem:
                                mov     cl,56
039E B138                       call    bdos
03A0 E87AFF     031D            ;       inc     al
                                ;       jz      memerr
                                NOP
03A3 90                         NOP
03A4 90                         NOP
03A5 90                         NOP
03A6 90                         ret
03A7 C3                         memerr:
                                mov     si,offset memm
03A8 BE8601                     jmps    errm
03AB EB11       03BE            ;
                                freemem:
                                mov     cl,57
03AD B139                       jmp     bdos
03AF E96BFF     031D            ;
                                load:
                                mov     cl,59
03B2 B138                       call    bdos
03B4 E866FF     031D            inc     ax              ;test for 0ffffh returned
03B7 40                         jz      loaderr
03B8 7401       03BB            ret
03BA C3                         loaderr:
                                mov     si,offset loadm
03BB BE9701                     ;
                                errm:
                                call    printm
03BE E87C00     043D            jmp     start
03C1 E9E3FE     02A7            ;
                                plmconout:
                                push    bp
03C4 55                         mov     bp,sp
03C5 8BEC                       mov     ax,4[bp]
03C7 8B4604                     call    conout
03CA E859FF     0326            pop     bp
03CD 5D                         ret     2
03CE C20200                     ;
                                plmgetline:
                                push    bp
03D1 55                         call    getline
03D2 E80200     03D7            pop     bp              ;restore bp for pl/m
03D5 5D                         ret
03D6 C3                         ;
                                ;       ************************************************
                                ;       *                                              *
                                ;       *    c o n s o l e   i / o   r o u t i n e s   *
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                        ;       *                                               *
                        ;       ************************************************
                        ;
                                if      debug
                        ;
                        ctlx:
                                mov     al,'*'
                                call    conout
                                call    crlf
                        getline:
                                mov     conptr,0
                        get0:
                                call    consin
                                cmp     al,3
                                jz      ctlc
                                cmp     al,8
                                jz      backsp
                                cmp     al,24           ;ctl-x
                                jz      ctlx
                                cmp     al,cr
                                jz      getlinedone
                                cmp     conptr,conbuffmax
                                jnb     getlinedone
                                mov     di,offset conbuff       ;normal character store
                                add     di,conptr
                                mov     [di],al
                                inc     conptr
                                jmps    get0
                        getlinedone:
                                mov     di,offset conbuff
                                add     di,conptr
                                mov     byte ptr [di],eol
                                mov     conptr,0
                                ret
                        backsp:
                                cmp     conptr,0
                                jz      get0
                                dec     conptr
                                call    blank
                                mov     al,8
                                call    conout
                                jmps    get0
                        ctlc:
                                mov     cl,0
                                mov     dl,0
                                jmp     bdos
                        ;
                                endif
                        ;
                                if      not debug
                        ;
                        getline:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                mov     dx,offset conbuffhdr
03D7 BAC111                     call    rdconbuff
03DA E851FF     032E            mov     bl,conbuffcnt
03DD 8A1EC211                   mov     bh,0
03E1 B700                       add     bx,offset conbuff
03E3 81C3C311                   mov     byte ptr [bx], eol
03E7 C6070D                     mov     conptr,0
03EA C70604120000               ret
03F0 C3
                        ;
                                endif
                        ;
                        conin:
03F1 BEC311                     mov     si,offset conbuff
03F4 03360412                   add     si,conptr
03F8 AC                         lods    al
03F9 FF060412                   inc     conptr
                                                        ;fall thru to upper
03FD 3C61               upper:  cmp     al,'a'          ;less than 'a'
03FF 7206       0407            jb      upret           ;or
0401 3C7A                       cmp     al,'z'          ;greater than 'z'
0403 7702       0407            ja      upret           ;then no change
0405 245F                       and     al,5fh          ;else convert to uc
0407 C3                 upret:  ret
                        ;
                        ctlchek:                        ;check for ctl-s, ctl-q and ctl-c
0408 E829FF     0334            call    constat         ;keypress?
040B 0AC0                       or      al,al           ;zero?
040D 740D       041C            jz      ctlexit         ;no keypress so return
040F E80EFF     0320            call    consin          ;if keypress then get the data
0412 3C13                       cmp     al,ctls         ;check for ctl-s
0414 7403       0419            jz      kwait
0416 E98EFE     02A7            jmp     start           ;any other key will restart
0419 E804FF     0320 kwait:     call    consin          ;if ctl-s then wait for another keypress
                        ctlexit:
041C C3                         ret
                        ;
                        crlf:
041D B00D                       mov     al,cr           ;send cr and lf to console
041F E804FF     0326            call    conout
0422 B00A                       mov     al,lf
0424 E8FFFE     0326            call    conout
0427 C3                         ret
                        ;
                        CRLFCHK:                        ;DO CRLF AND CHECK FOR ABORT
0428 E8F2FF     041D            CALL    CRLF
042B E8DAFF     0408            CALL    CTLCHEK
042E C3                         RET
                        ;
                        blank:                          ;print a blank.
042F B020                       mov     al,' '
0431 E8F2FE     0326            call    conout
0434 C3                         ret
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                        ;
                        tabs:                   ;print # blanks in cx
0435 51                         push    cx
0436 E8F6FF     042F            call    blank
0439 59                         pop     cx
043A E2F9       0435            loop    tabs
043C C3                         ret
                        ;
                        printa:                 ;print the message at [si] on console
                                                ;end of message indicated by parity set.
043D AC                         lods    al      ;get a byte
043E A880                       test    al,80h  ;check for end of message
0440 7507       0449            jnz     pquit   ;quit if parity set
0442 56                         push    si
0443 E8E0FE     0326            call    conout  ;otherwise display byte
0446 5E                         pop     si
0447 EBF4       043D            jmps    printa  ;print more message
                        pquit:
0449 247F                       and     al,7fh  ;strip parity
044B E8D8FE     0326            call    conout  ;print last byte
044E C3                         ret
                        ;
                        ascout:                 ;output [al] in ascii
044F 3C20                       cmp     al,' '
0451 7204       0457            jb      perout  ;less than blank?
0453 3C7E                       cmp     al,7eh
0455 7602       0459            jna     ascend
                        perout:
0457 B02E                       mov     al,'.'  ;output '.'
                        ascend:
0459 E8CAFE     0326            call    conout  ;else output ascii
045C C3                         ret
                        ;
                        print8or16:             ;print byte or word at es:[si] depending on wmode
045D 268B04                     mov     ax,es:[si]
0460 F6066D1201                 test    wmode,1
0465 7415       047C            jz      printbyte
0467 EB0C       0475            jmps    printword
                        ;
                        printdword:             ;print double word as: ssss:oooo
                                ; called with:
                                ;  es = segment
                                ;  di = offset
0469 57                         push    di
046A 8CC0                       mov     ax,es
046C E80600     0475            call    printword ;print segment
046F B03A                       mov     al,':'
0471 E8B2FE     0326            call    conout
0474 58                         pop     ax
                        ;
                        printword:              ;print value in [ax] as 4 hex digits
0475 50                         push    ax
```

CP/M-86 v.1.1 (Vol. III)
COPYRIGHT (C) 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
C81-162

```
690
691  0476 8AC4                  mov     al,ah
692  0478 E80100       047C     call    printbyte
693  047B 58                    pop     ax
694                        ;
695                        printbyte:                  ;print value in [al] as 2 hex digits
696  047C 50                    push    ax
697  047D B104                  mov     cl,4
698  047F D2E8                  shr     al,cl           ;shift al right 4 places
699  0481 E80100       0485     call    printnibble     ;output upper nibble
700  0484 58                    pop     ax              ;restore al (now we do lower nibble)
701                        ;
702                        printnibble:                ;print value in low 4 bits of [al] as a hex digit
703  0485 240F                  and     al,0fh          ;mask upper 4 bits
704  0487 0490                  add     al,90h
705  0489 27                    daa
706  048A 1440                  adc     al,40h
707  048C 27                    daa
708  048D E896FE       0326     call    conout
709  0490 C3                    ret
710                        ;
711                        ;       ****************************************
712                        ;       *                                      *
713                        ;       *     file name parsing routines       *
714                        ;       *                                      *
715                        ;       ****************************************
716                        ;
717                        parse:                      ;parse into fcb whose offset is in [di]
718  0491 0E                    push    cs
719  0492 07                    pop     es              ;set es=cs
720  0493 57                    push    di              ;save fcb address
721  0494 2AC0                  sub     al,al
722  0496 B92400                mov     cx,36           ;fcblen
723  0499 F3AA          rep     stos    al              ;initialize fcb to 0
724  049B 5F                    pop     di              ;restore fcb address
725                        ;
726                        parse2:                     ;enter here to parse without clearing
727                                                    ;assumes es = cs from parse:
728  049C 893E5D00              mov     fcbadr,di       ;save fcb address
729  04A0 47                    inc     di              ;point to first byte of filename
730  04A1 E82200       04C6     call    setupdisk       ;check for d: and set drive byte
731  04A4 B90800                mov     cx,8
732  04A7 E84B00       04F5     call    fillfield       ;first item was disk, now get filename
733  04AA B90300                mov     cx,3            ;length of file type
734  04AD 803E5D0012E           cmp     lastchar,'.'
735  04B2 7405         04B9     jz      filltype
736  04B4 E86000       0517     call    fillbl          ;fill type with blanks if no '.'
737  04B7 EB03         04BC     jmps    parseret
738                        filltype:
739  04B9 E83900       04F5     call    fillfield       ;if '.', fill field from console buff
740                        parseret:
741  04BC E85F00       051E     call    scanq           ;count '?'s in fcb
742  04BF A05D12                mov     al,lastchar
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
USER # _____

```
                                ret                     ;with last char scanned in [al]
                        parseerr:
04C3 E919FE     02DF            jmp     err
                        ;
                        setupdisk:                      ;set byte 0 of fcb according to char in fcb (1)
04C6 E823FF     03F1            call    conin
04C9 A2AD12                     mov     lastchar,al
04CC 3C20                       cmp     al,' '
04CE 74F6       04C6            jz      setupdisk       ;a blank input
04D0 3C0D                       cmp     al,eol
04D2 7413       04E7            jz      s1              ;can't be drive, decrement conptr to rescan
04D4 E81AFF     03F1            call    conin
04D7 3C3A                       cmp     al,':'
04D9 7508       04E3            jnz     s0              ;not a drive, subtract 2 from conptr to rescan
04DB A0AD12                     mov     al,lastchar     ;get drive char
04DE 2C40                       sub     al,'A'-1
04E0 4F                         dec     di              ;point to fcb (0)
04E1 AA                         stos    al              ;store drive byte
04E2 C3                         ret
                        s0:
04E3 FF0E0412                   dec     conptr
                        s1:
04E7 FF0E0412                   dec     conptr
04EB C3                         ret
                        ;
                        pdelim:                         ;check char in [al] for delimiter; return ZF if so.
04EC BFA212                     mov     di,offset delims
04EF B90B00                     mov     cx,ndelims
04F2 F2AE               repnz   scas    al              ;look in table
04F4 C3                         ret
                        ;
                        fillfield:                      ;count in [cx], dest ptr in [di]
04F5 E8F9FE     03F1            call    conin
04F8 A2AD12                     mov     lastchar,al     ;save last char scanned
04FB 3C2A                       cmp     al,'*'
04FD 7505       0504            jnz     notast
04FF E81100     0513            call    fillq           ;fill with '?'
0502 EBF1       04F5            jmps    fillfield       ;continue till delimiter
                        notast:
0504 57                         push    di
0505 51                         push    cx
0506 E8E3FF     04EC            call    pdelim
0509 59                         pop     cx
050A 5F                         pop     di
050B 740A       0517            jz      fillbl          ;if delimiter, fill field with ' '
050D E3B4       04C3            jcxz    parseerr        ;error if count exceeded
050F AA                         stos    al              ;store char in fcb
0510 49                         dec     cx              ;decrement count
0511 E8E2       04F5            jmps    fillfield
                        fillq:
0513 B03F                       mov     al,'?'
0515 EB02       0519            jmps    fillx
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```
                                fillbl:
0517 B020                               mov     al,' '
                                fillx:
0519 E302               051D            jcxz    filldone
051B F3AA                       rep     stos    al              ;store '?' or ' '
                                filldone:
051D C3                                 ret
                                ;
                                scanq:                          ;count '?'s in fcb, return ZF set if found
051E 8B3EAE12                           mov     di,fcbadr
0522 47                                 inc     di              ;point to first char of filename
0523 B90B00                             mov     cx,11           ;11 chars to check
0526 B03F                               mov     al,'?'
0528 F2AE                       repnz   scas    al
052A C3                                 ret
                                ;
                                ;       ************************************
                                ;       *                                  *
                                ;       *     user CPU state routines      *
                                ;       *                                  *
                                ;       ************************************
                                ;
                                chkreg:                         ;if reg name in [ax] is valid, return with
                                                                ;register number in regnum
                                                                ;else go to error processor
052B B90E00                             mov     cx,totreg+1     ;number of names to check + 1
052E BF4D01                             mov     di,offset regname
0531 0E                                 push    cs
0532 07                                 pop     es
0533 F2AF                       repnz   scas    ax
0535 E320               0557            jcxz    checkerr        ;not a valid reg name
0537 BA0D00                             mov     dx,totreg
053A 2BD1                               sub     dx,cx
053C 89165512                           mov     regnum,dx       ;save reg number
0540 C3                                 ret
                                ;
                                checkflag:                      ;check for valid flag name
0541 B90A00                             mov     cx,nflag+1      ;number of names to check + 1
0544 BF6701                             mov     di,offset flagname
0547 0E                                 push    cs
0548 07                                 pop     es
0549 F2AE                       repnz   scas    al
054B E30A               0557            jcxz    checkerr        ;not a valid flag name
054D BA0900                             mov     dx,nflag
0550 2BD1                               sub     dx,cx
0552 89165512                           mov     regnum,dx       ;save flag number
0556 C3                                 ret
                                ;
                                checkerr:
0557 E985FD             02DF            jmp     err
                                ;
                                setreg:                         ;set reg whose number is in [cx] to value in [ax]
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                                ;
055A BE9C11                     mov     si,offset userreg
055D 03F1                       add     si,cx
055F 03F1                       add     si,cx
0561 8904                       mov     [si],ax
0563 C3                         ret
                                ;
                        printflags:                     ;print values of flags
0564 B90000                     mov     cx,0
                        pf0:
0567 51                         push    cx              ;save count
0568 E84B00     05B6            call    printflag
056B 59                         pop     cx
056C 41                         inc     cx
056D 83F909                     cmp     cx,9
0570 72F5       0567            jb      pf0
0572 C3                         ret
                                ;
                        setflag:                        ;set flag whose # is in [cx] to value in [bx]
0573 BE7001                     mov     si,offset flagbits
0576 03F1                       add     si,cx
0578 AC                         lods    al
0579 8AC8                       mov     cl,al
057B B8FEFF                     mov     ax,0fffeh
057E D3C8                       ror     ax,cl
0580 D3CB                       ror     bx,cl
0582 2E2106B611                 and     userfl,ax
0587 2E091EB611                 or      userfl,bx
058C C3                         ret
                                ;
                        printflagname:                  ;print flag name whose # is in [cx]
058D BE6701                     mov     si,offset flagname
0590 03F1                       add     si,cx
0592 AC                         lods    al              ;get flag name
0593 E890FD     0326            call    conout
0596 C3                         ret
                                ;
                        getflag:                        ;check flag whose # is in [cx]
                                                        ;return with ZF set if flag is set
0597 B80100                     mov     ax,1
059A BE7001                     mov     si,offset flagbits
059D 03F1                       add     si,cx
059F 8A0C                       mov     cl,[si]         ;get flagbits (flagnum)
05A1 D3C8                       ror     ax,cl           ;get mask into position
05A3 2E2306B611                 and     ax,userfl       ;see if bit set in CPU state
05A8 C3                         ret
                                ;
                        printflagval:                   ;print value of flag (as 0 or 1) whose # is in [cx]
05A9 E8EBFF     0597            call    getflag
05AC B030                       mov     al,'0'
05AE 7402       05B2            jz      pf2             ;if flag not set, print '0'
05B0 B031                       mov     al,'1'          ;else print '1'
                        pf2:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
 902
 903  05B2 E871FD      0326          call    conout
 904  05B5 C3                        ret
 905                            ;
 906                            printflag:                      ;print flag (as flagname of '-') whose # is in [cx]
 907  05B6 BE6701                       mov     si,offset flagname
 908  05B9 03F1                         add     si,cx           ;point to flag name
 909  05BB FF34                         push    word ptr [si]   ;save flag char
 910  05BD E8D7FF      0597             call    getflag
 911  05C0 58                           pop     ax              ;get flag char
 912  05C1 7502        05C5             jnz     pname           ;if flag set, use flag char
 913  05C3 B02D                         mov     al,'-'          ;else print hyphen
 914                            pname:
 915  05C5 E85EFD      0326             call    conout
 916  05C8 C3                           ret
 917                            ;
 918                            PREG1:                          ;PRINT FIRST 6 REGISTER NAMES (FOR 40 COLUMNS)
 919  05C9 B90000                       MOV     CX,0
 920  05CC C606531206                   MOV     NREG,6
 921  05D1 EB0D        05E0             JMPS    PR0
 922                            PREG2:                          ;PRINT NEXT 7 REGISTER NAMES (FOR 40 COLUMNS)
 923  05D3 B90600                       MOV     CX,6
 924  05D6 EB03        05DB             JMPS    PR00
 925                            printregs:                      ;print register values
 926  05D8 B90000                       mov     cx,0
 927                            PR00:
 928  05DB C606531200                   MOV     NREG,13
 929                            pr0:
 930  05E0 E82900      060C             call    testregcl       ;see if reg should be printed
 931  05E3 7308        05ED             jnb     pr2             ;don't print if carry not set
 932  05E5 51                           push    cx
 933  05E6 E80C00      05F5             call    printregval
 934  05E9 E843FE      042F             call    blank
 935  05EC 59                           pop     cx
 936                            pr2:
 937  05ED 41                           inc     cx
 938  05EE 3A0E5312                     CMP     CL,NREG
 939  05F2 72EC        05E0             jb      pr0
 940  05F4 C3                           ret
 941                            ;
 942                            printregval:                    ;print value of reg whose # is in [cx]
 943  05F5 BE7C11                       mov     si,offset userreg
 944  05F8 03F1                         add     si,cx
 945  05FA 03F1                         add     si,cx
 946  05FC AD                           lods    ax
 947  05FD E875FE      0475             call    printword
 948  0600 C3                           ret
 949                            ;
 950                            printregname:                   ;print name of reg whose # is in [cx]
 951  0601 BE4D01                       mov     si,offset regname
 952  0604 03F1                         add     si,cx
 953  0606 03F1                         add     si,cx
 954  0608 E832FE      043D             call    printm
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
 955
 956  060B C3                        ret
 957                          ;
 958                          testregcl:                  ;see if reg whose # is in [cl] should be printed
 959                                                      ;return with carry set if so
 960  060C F6065412FF                 test    segflag,0ffh
 961  0611 7509          061C         jnz     printit     ;print all reg's if segflag set
 962  0613 80F90B                     cmp     cl,11       ;otherwise, see if [cl] has seg reg #
 963  0616 7704          061C         ja      printit
 964  0618 80F908                     cmp     cl,8
 965  061B C3                         ret
 966                          printit:
 967  061C F9                         stc
 968  061D C3                         ret
 969                          ;
 970                          SETUPHDR:
 971  061E E8FCFD        041D         call    crlf
 972  0621 B90B00                     mov     cx,11
 973  0624 E80EFE        0435         call    tabs        ;leave space for flags
 974  0627 B90000                     mov     cx,0
 975  062A C3                         RET
 976                          ;
 977                          PREGHDR1:                   ;PRINT HEADER FOR FIRST 6 REGS (FOR 40 COL)
 978  062B E8F0FF        061E         CALL    SETUPHDR
 979  062E C606531206                 MOV     NREG,6
 980  0633 EB10          0645         JMPS    PRH0
 981                          PREGHDR2:                   ;PRINT HEADER FOR NEXT 7 REGISTERS
 982  0635 E8F7FD        042F         CALL    BLANK
 983  0638 B90600                     MOV     CX,6
 984  063B E303          0640         JMPS    PRH00
 985                          printregheader:             ;print header for registers
 986  063D E8DEFF        061E         CALL    SETUPHDR
 987                          PRH00:
 988  0640 C606531200                 MOV     NREG,13
 989                          prh0:
 990  0645 E8C4FF        060C         call    testregcl   ;see if reg should be printed
 991  0648 730B          0655         jnb     prh1        ;don't print if carry not set
 992  064A 51                         push    cx
 993  064B E8B3FF        0601         call    printregname
 994  064E B90300                     mov     cx,3
 995  0651 E8E1FD        0435         call    tabs
 996  0654 59                         pop     cx
 997                          prh1:
 998  0655 41                         inc     cx
 999  0656 3A0E5312                   CMP     CL,NREG
1000  065A 72E9          0645         jb      prh0
1001  065C C3                         ret
1002                          ;
1003                          printinstr:                 ;disassemble instruction at [cs:ip]
1004  065D 2E8E06AC11                 mov     es,usercs
1005  0662 2E8B36B411                 mov     si,userip
1006  0667 F6063912FF                 test    disempresent,0ffh
1007  066C 7410          067E         jz      pil
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1008
1009  066E 06                        push    es
1010  066F 56                        push    si
1011  0670 F6065412FF                test    segflag,0ffh
1012  0675 7403          067A        jz      pi0
1013  0677 E8A3FD        041D        call    crlf
1014                           pi0:
1015  067A E8830D        1400        call    disem
1016  067D C3                        ret
1017                           pi1:
1018  067E 268A04                    mov     al,es:[si]
1019  0681 E8F8FD        047C        call    printbyte
1020  0684 C3                        ret
1021                           ;
1022                           printseginfo:                   ;print name, start and end address of segment whose
1023                                                           ;number is in [al] if length is non-zero.
1024  0685 8AC8                      mov     cl,al           ;save seg number
1025  0687 B306                      mov     bl,6
1026  0689 F6E3                      mul     bl              ;6 bytes per segment in base page
1027  068B 057212                    add     ax,offset basepagesave
1028  068E 8BF0                      mov     si,ax           ;si now points to entry in base page
1029  0690 AD                        lods    ax              ;get low 16 bits of length
1030  0691 8BD8                      mov     bx,ax           ;save in bx
1031  0693 AC                        lods    al              ;get high nibble of length
1032  0694 8AD0                      mov     dl,al           ;save it
1033  0696 B400                      mov     ah,0
1034  0698 0BC3                      or      ax,bx           ;test for zero length
1035  069A 7430          06CC        jz      psiret          ;if zero, no display
1036  069C AD                        lods    ax              ;get base
1037  069D 53                        push    bx              ;save low (length)
1038  069E 52                        push    dx              ;save high (length)
1039  069F 50                        push    ax              ;save base
1040  06A0 B500                      mov     ch,0            ;zero high byte of segment #
1041  06A2 D1E1                      shl     cx,1            ;* 2 (2 bytes per segment name)
1042  06A4 81C17901                  add     cx,offset segnames      ;cx now points to segment name
1043  06A8 51                        push    cx              ;save it
1044  06A9 E871FD        041D        call    crlf
1045  06AC 5E                        pop     si
1046  06AD E88DFD        043D        call    printm          ;print segment name
1047  06B0 E87CFD        042F        call    blank
1048  06B3 07                        pop     es              ;get base
1049  06B4 06                        push    es              ;save base
1050  06B5 BF0000                    mov     di,0
1051  06B8 E8AEFD        0469        call    printdword      ;print start address
1052  06BB E871FD        042F        call    blank
1053  06BE 5B                        pop     bx              ;get base
1054  06BF 58                        pop     ax              ;get high (len)
1055  06C0 B10C                      mov     cl,12
1056  06C2 D3E0                      shl     ax,cl           ;move ls nibble of al to ms nibble of ah
1057  06C4 03C3                      add     ax,bx           ;add ms nibble of length to base
1058  06C6 8EC0                      mov     es,ax
1059  06C8 5F                        pop     di              ;get low (len)
1060  06C9 E89DFD        0469        call    printdword      ;print end address
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1061
1062                            psiret:
1063   06CC C3                          ret
1064                            ;
1065                            setcpustate:                    ;set users regs after E command
1066   06CD A1BD11                      MOV     AX,CCPSS
1067   06D0 2EA3B011                    MOV     USERSS,AX
1068   06D4 A1BF11                      MOV     AX,CCPSP
1069   06D7 2EA3A411                    MOV     USERSP,AX       ;RESET USER STACK POINTER TO CCP STACK
1070   06DB 8BC3                        mov     ax,bx           ;get ds base in [ax] (returned from bdos load)
1071   06DD 2EA3AE11                    mov     userds,ax       ;set user's ds
1072   06E1 2EA3AC11                    mov     usercs,ax
1073   06E5 2EA3B211                    mov     useres,ax
1074   06E9 A35A12                      mov     typelseg,ax
1075   06EC A35E12                      mov     type2seg,ax
1076   06EF 8EC0                        mov     es,ax
1077   06F1 26F606050001                test    es:byte ptr .5, 1       ;test 8080 flag
1078   06F7 7405          06FE          jz      not8080
1079   06F9 B80001                      mov     ax,100h         ;default for userip, lasloc, disloc
1080   06FC EB2D          072B          jmps    setdone
1081                            not8080:
1082   06FE 26A10300                    mov     ax,es:.3        ;get cs base
1083   0702 2EA3AC11                    mov     usercs,ax
1084   0706 A35A12                      mov     typelseg,ax
1085   0709 26A10F00                    mov     ax,es:.15       ;get es base
1086   070D 0BC0                        or      ax,ax
1087   070F 7404          0715          jz      sc1
1088   0711 2EA3B211                    mov     useres,ax       ;set it if there was one
1089                            sc1:
1090   0715 26A11500                    mov     ax,es:.21       ;get ss base
1091   0719 0BC0                        or      ax,ax
1092   071B 740E          072B          jz      setdone
1093   071D 2EA3B011                    mov     userss,ax       ;set it if there was one
1094   0721 26A11200                    mov     ax,es:.18       ;get stack length
1095   0725 2EA3A411                    mov     usersp,ax       ;set user's sp
1096   0729 2BC0                        sub     ax,ax           ;userip, lasloc, disloc = 0 for non-8080 model
1097                            setdone:
1098   072B 2EA3B411                    mov     userip,ax
1099   072F A33112                      mov     lasloc,ax
1100   0732 C3                          ret
1101                            ;
1102                            ;       ***********************************************
1103                            ;       *                                             *
1104                            ;       *     breakpoint/single step procedures       *
1105                            ;       *                                             *
1106                            ;       ***********************************************
1107                            ;
1108                            setbp:                          ;set breakpoint at address stored in dword at [di]
1109   0733 C435                        les     si,[di]         ;point [es]:[si] to breakpoint location
1110   0735 B0CC                        mov     al,0cch         ;int 3 instruction
1111   0737 268604                      xchg    al,es:[si]      ;set breakpoint and fetch instruction
1112   073A 884504                      mov     4[di],al        ;save user instruction in breakpoint table
1113   073D FE062612                    inc     bpcnt           ;increment breakpoint count
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1114
1115  0741 C3                    ret
1116                        ;
1117                        bpclear:                        ;clear breakpoint table
1118  0742 2BC9                    sub     cx,cx
1119  0744 8A0E2612                mov     cl,bpcnt        ;get # of bp's to clear
1120  0748 BE2712                  mov     si,offset brkiloc       ;point to bp table
1121                        bpcloop:
1122  074B E30F          075C      jcxz    bpend           ;0..quit
1123  074D AD                      lods    ax              ;get bp offset
1124  074E 50                      push    ax              ;save it
1125  074F AD                      lods    ax              ;get bp segment
1126  0750 8EC0                    mov     es,ax
1127  0752 AC                      lods    al              ;get inst byte
1128  0753 5F                      pop     di              ;get bp offset
1129  0754 AA                      stos    al              ;store user instruction back
1130  0755 E2F4          074B      loop    bpcloop
1131  0757 C606261200              mov     bpcnt,0         ;zero bp counter
1132  075C C3                bpend:  ret
1133                        ;
1134                        BPV:
1135  075D F6066F12FF              TEST    SAVEVECFLAG,0FFH
1136  0762 741C          0780      JZ      BPVECTRET
1137                        bpvect:                         ;set up breakpoint/single step vectors
1138  0764 E81A00        0781      call    savevect
1139  0767 BA0000                  mov     dx,0
1140  076A 8EC2                    mov     es,dx           ;make sure dest is absolute 0
1141  076C BF0400                  mov     di,4            ;set up single step vector
1142  076F B8DF07                  mov     ax,offset ssentry       ;single step entry point
1143  0772 AB                      stos    ax              ;save at ss vector
1144  0773 8CC8                    mov     ax,cs
1145  0775 AB                      stos    ax              ;save cs
1146  0776 BF0C00                  mov     di,12           ;set up breakpoint vector
1147  0779 B8D707                  mov     ax,offset breakentry    ;set up bp vector
1148  077C AB                      stos    ax              ;save at bp vector
1149  077D 8CC8                    mov     ax,cs
1150  077F AB                      stos    ax              ;save cs
1151                        BPVECTRET:
1152  0780 C3                      ret
1153                        ;
1154                        savevect:                       ;save previous contents of 0:4 thru 0:f
1155  0781 BE0400                  mov     si,4
1156  0784 BF1A12                  mov     di,offset vectorsave
1157  0787 1E                      push    ds
1158  0788 07                      pop     es              ;point to ddt segment
1159  0789 B90C00                  mov     cx,12
1160  078C 1E                      push    ds
1161  078D BA0000                  mov     dx,0
1162  0790 8EDA                    mov     ds,dx
1163  0792 F3A4          rep       movs    al,al
1164  0794 1F                      pop     ds
1165                        svret:
1166  0795 C3                      ret
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1167
1168                            ;
1169                            restorevect:                    ;restore previous contents of 0:4 thru 0:f
1170  0796 F6066F12FF                   test    savevecflag,0ffh
1171  079B 7410         07AD            jz      rvret
1172  079D BE1A12                       mov     si,offset vectorsave
1173  07A0 BF0400                       mov     di,4
1174  07A3 B9CC00                       mov     cx,12
1175  07A6 BA0000                       mov     dx,0
1176  07A9 8EC2                         mov     es,dx
1177  07AB F3A4                 rep     movs    al,al
1178                            rvret:
1179  07AD C3                           ret
1180                            ;
1181                            SETDEFSEG:                      ;SET DEFAULT TYPE1SEG, LASLOC
1182  07AE 2E8B3EB411                   mov     di,userip
1183  07B3 893E3112                     mov     lasloc,di       ;set disassembler offset
1184  07B7 2E8E06AC11                   mov     es,usercs
1185  07BC 8C065A12                     mov     type1seg,es     ;set type1seg segment
1186  07C0 C3                           RET
1187                            ;
1188                            breakaddr:                      ;print address where break occurred
1189  07C1 E859FC       041D            call    crlf
1190  07C4 B02A                         mov     al,'*'
1191  07C6 E85DFB       0326            call    conout
1192  07C9 2EA1AE11                     mov     ax,userds
1193  07CD A35E12                       mov     type2seg,ax     ;set type2 segment to userds
1194  07D0 E8DBFF       07AE            CALL    SETDEFSEG
1195  07D3 E893FC       0469            call    printdword      ;print break address
1196  07D6 C3                           ret
1197                            ;
1198                            breakentry:                     ;breakpoint entry
1199  07D7 2EC606BC1101                 mov     breakf1,1
1200  07DD EB06         07E5            jmps    savecpu         ;common code for break and single step
1201                            ssentry:                        ;single step entry
1202  07DF 2EC606BC1100                 mov     breakf1,0
1203                            savecpu:
1204  07E5 2EA39C11                     mov     userax,ax
1205  07E9 2E891E9E11                   mov     userbx,bx
1206  07EE 2E890EA011                   mov     usercx,cx
1207  07F3 2E8916A211                   mov     userdx,dx
1208  07F8 2E8936A811                   mov     usersi,si
1209  07FD 2E893EAA11                   mov     userdi,di
1210  0802 2E892EA611                   mov     userbp,bp
1211  0807 2E8926A411                   mov     usersp,sp
1212  080C 2E8C06B211                   mov     useres,es
1213  0811 2E8C1EAE11                   mov     userds,ds
1214  0816 2E8C16B011                   mov     userss,ss
1215  081B 8BEC                         mov     bp,sp
1216  081D 8B4600                       mov     ax,[bp]
1217  0820 2EA3B411                     mov     userip,ax
1218  0824 8B4602                       mov     ax,2[bp]
1219  0827 2EA3AC11                     mov     usercs,ax
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1220
1221  082B 8B4604                    mov     ax,4[bp]
1222  082E 2EA3B611                  mov     userfl,ax
1223                          ;
1224  0832 8CC3                      mov     ax,cs
1225  0834 8ED8                      mov     ds,ax
1226  0836 8ED0                      mov     ss,ax
1227  0838 BC1013                    mov     sp,offset stackp
1228                          ;
1229  083B F6066B1202                test    sysif,ifmask8   ;see whether interrupts should be enable
1230  0840 7401          0843        jz      sav0
1231  0842 FB                        sti
1232                          sav0:
1233  0843 2E8306A41106              add     usersp,6        ;to make up for stacked cs, ip, and fl
1234  0849 FC                        cld
1235  084A E849FF        0796        call    restorevect
1236  084D 2EF606BC11FF              test    breakfl,0ffh
1237  0853 741A          086F        jz      sst0
1238                          ;
1239                          break0:                         ;continuation of break processing
1240  0855 2EFF0EB411                dec     userip          ;adjust user ip for breakpoint instr
1241  085A E8E5FE        0742        call    bpclear         ;clear all breakpoints
1242  085D F6064912FF                test    skipbdos,0ffh   ;were we originally in trace mode?
1243  0862 7506          086A        jnz     sst00           ;if so, continue tracing
1244                          ;
1245                          tracedone:
1246  0864 E85AFF        07C1        call    breakaddr
1247  0867 E93DFA        02A7        jmp     start
1248                          ;
1249                          sst00:                          ;get here on breakpoint on return from bdos
1250  086A C606491200                mov     skipbdos,0      ;no longer tracing thru bdos
1251                          sst0:                           ;continuation of single step processing
1252  086F 2E8126B611FF              and     userfl,0feffh   ;clear user trap flag
1253      FE
1254  0876 F6064A1201                TEST    USERIFOFF,1
1255  087B 740C          0889        JZ      SST1
1256  087D C6064A1200                MOV     USERIFOFF,0
1257  0882 2E810EB61100              OR      USERFL,200H     ;RESTORE USER IF
1258      02
1259                          SST1:
1260  0889 FF0E4612                  dec     tracecount
1261  088D 74D5          0864        jz      tracedone
1262  088F E876FB        0408        CALL    CTLCHEK         ;CHECK FOR ABORT
1263  0892 F6064812FF                test    traceprint,0ffh
1264  0897 7403          089C        jz      tracerestore
1265  0899 E80608        10A2        call    xnohdr          ;display regs without header
1266                          ;
1267                          tracerestore:                   ;enter here when in trace mode
1268  089C 2E8E06AC11                mov     es,usercs
1269  08A1 2E8B36B411                mov     si,userip
1270  08A6 A11201                    mov     ax,word ptr .bdosintloc ;get bdos interrupt instruction
1271  08A9 263B04                    cmp     ax,es:[si]      ;see if instruction to be traced is bdos int
1272  08AC 7518          08C6        jnz     tr00
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1273
1274  08AE 8C062912              mov   brk1seg,es
1275  08B2 83C602                add   si,2          ;point to instruction after bdos int
1276  08B5 89362712              mov   brk1loc,si
1277  08B9 BF2712                mov   di,offset brk1loc
1278  08BC E874FE      0733      call  setbp         ;set breakpoint at return from bdos
1279  08BF C606491201            mov   skipbdos,1    ;so we know we were in trace mode when we hit bp
1280  08C4 EB39        08FF      jmps  rstore        ;without setting single step flag
1281                         tr00:
1282  08C6 2EA1B611              MOV   AX,USERFL
1283  08CA 0D0001                OR    AX,100H       ;SET TRACE FLAG
1284  08CD A90002                TEST  AX,200H       ;IS USER IF SET?
1285  08D0 7429        08FB      JZ    TR01
1286  08D2 25FFFD                AND   AX,NOT 200H   ;CLEAR IT (SO WE DON'T END UP IN INT HANDLER)
1287  08D5 2E8E06AC11            MOV   ES,USERCS
1288  08DA 2E8B36B411            MOV   SI,USERIP
1289  08DF 268A1C                MOV   BL,ES:[SI]    ;GET INSTRUCTION TO EXECUTE
1290  08E2 80FBFA                CMP   BL,0FAH       ;IS IT CLI?
1291  08E5 7414        08FB      JZ    TR01
1292  08E7 80FBCF                CMP   BL,0CFH       ;IRET?
1293  08EA 740F        08FB      JZ    TR01
1294  08EC 80FB9D                CMP   BL,09DH       ;POPF?
1295  08EF 740A        08FB      JZ    TR01
1296  08F1 80FBCD                CMP   BL,0CDH       ;INT?
1297  08F4 7405        08FB      JZ    TR01
1298  08F6 C6064A1201            MOV   USERIFOFF,1   ;SET FLAG SO DDT86 WILL TURN IF BACK ON
1299                         TR01:
1300  08FB 2EA3B611              MOV   USERFL,AX
1301                         rstore:                 ;enter here when in G mode
1302  08FF E85BFE      075D      CALL  BPV
1303  0902 FA                    cli
1304  0903 BC9C11                mov   sp,offset userreg     ;point to reg save area
1305  0906 58                    pop   ax
1306  0907 5B                    pop   bx
1307  0908 59                    pop   cx
1308  0909 5A                    pop   dx
1309  090A 2E8F06BA11            pop   savesp
1310  090F 5D                    pop   bp
1311  0910 5E                    pop   si
1312  0911 5F                    pop   di
1313  0912 1F                    pop   ds            ;throw away cs
1314  0913 1F                    pop   ds
1315  0914 2E8F06B811            pop   savess
1316  0919 07                    pop   es
1317  091A 2E8E16B811            mov   ss,savess     ;restore stack
1318  091F 2E8B26BA11            mov   sp,savesp
1319  0924 2EFF36B611            push  userfl        ;flags
1320  0929 2EFF36AC11            push  usercs        ;cs
1321  092E 2EFF36B411            push  userip        ;ip
1322  0933 CF                    iret                ;transfer to user program
1323                         ;
1324                         ;     ************************************
1325                         ;     *                                  *
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1326
1327                        ;       *     miscellaneous routines     *
1328                        ;       *                                *
1329                        ;       **********************************
1330                        ;
1331                        delim:
1332  0934 3C0D                     cmp     al,eol
1333  0936 740A       0942          jz      delret
1334  0938 3C2C                     cmp     al,','
1335  093A 7406       0942          jz      delret
1336  093C 3C20                     cmp     al,' '
1337  093E 7402       0942          jz      delret
1338  0940 3C3A                     cmp     al,':'
1339                        delret:
1340  0942 C3                       ret
1341                        ;
1342                        hexcon:
1343  0943 2C30                     sub     al,'0'
1344  0945 3C0A                     cmp     al,10
1345  0947 720A       0953          jb      hexret
1346  0949 04F9                     add     al,('0' - 'A' + 10) and 0ffh
1347  094B 3C10                     cmp     al,16
1348  094D 7305       0954          jnb     hexerr
1349  094F 3C0A                     cmp     al,10
1350  0951 7201       0954          jb      hexerr
1351                        hexret:
1352  0953 C3                       ret
1353                        hexerr:
1354  0954 E938F9     02DF          jmp     err
1355                        ;
1356                        plmset:                         ;get here when assembler wants to set memory
1357  0957 8BEC                     mov     bp,sp           ;for parameter fetching
1358  0959 8B4602                   mov     ax,2[bp]        ;get value in [al]
1359  095C 8E065A12                 mov     es,typelseg     ;segment used in A command is in typel seg
1360  0960 8B7E04                   mov     di,4[bp]        ;get offset from stack
1361  0963 E81A00     0980          call    setbyte         ;set and verify
1362  0966 47                       inc     di              ;increment offset
1363  0967 7503       096C          jnz     psret           ;if incremented offset is non-zero, return
1364  0969 E93BF9     02A7          jmp     start           ;otherwise exit A command, since wrap occurred
1365                        psret:
1366  096C C20400                   ret     4               ;remove 2 parameters
1367                        ;
1368                        set8or16:                       ;set byte or word at es:[di] depending on wmode
1369  096F F6065D1201               test    wmode,1
1370  0974 740A       0980          jz      setbyte
1371                                                        ;fall through to setword
1372                        ;
1373                        setword:                        ;set word at es:[di] to [ax] and verify
1374                        ;
1375                        ;       NOTE:  THIS CODE COULD BE REPLACED BY THE FOLLOWING 4 INSTRUCTIONS
1376                        ;              FOR SYSTEMS IN WHICH MEMORY CAN ONLY BE ADDRESSED BY WORDS.
1377                        ;              HOWEVER, THIS WILL WRAP AROUND AND MODIFY LOCATIONS 0 IF
1378                        ;              [DI] CONTAINS OFFFEH.
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1379
1380                            ;
1381                            ;       MOV     ES:[DI],AX
1382                            ;       CMP     ES:[DI],AX
1383                            ;       JNZ     BADVER
1384                            ;       RET
1385                            ;
1386  0976 50                           push    ax              ;save hi byte
1387  0977 E80600       0980            call    setbyte         ;set low byte
1388  097A 5F                           pop     ax
1389  097B 8AC4                         mov     al,ah
1390  097D 47                           inc     di              ;point to next location
1391  097E 7408         0988            jz      sret            ;don't set byte if wraparound occurred
1392                                                              ;fall thru to setbyte
1393                            ;
1394                            setbyte:                        ;set byte at es:[di] to [al] and verify
1395  0980 263805                       mov     es:[di],al      ;store byte
1396  0983 263805                       cmp     es:[di],al      ;see if it got stored
1397  0986 7501         0989            jnz     badver
1398                            sret:
1399  0988 C3                           ret
1400                            ;
1401                            badver:
1402  0989 57                           push    di
1403  098A 06                           push    es
1404  098B BEE601                       mov     si,offset verm
1405  098E E8ACFA       043D            call    printa          ;print verify error message
1406  0991 07                           pop     es
1407  0992 5F                           pop     di
1408  0993 E8D3FA       0469            call    printdword
1409  0996 E90EF9       02A7            jmp     start
1410                            ;
1411                            inclor2:                        ;inc pointer at [si] by 1 or 2, depending on wmode
1412                                                            ;return with carry flag set if wrap occurred
1413  0999 A06D12                       mov     al,wmode
1414  099C 250100                       and     ax,1            ;mask to 0 or 1
1415  099F 40                           inc     ax              ;increment value is now 1 or 2
1416  09A0 0104                         add     [si],ax         ;increment pointer
1417  09A2 3904                         cmp     [si],ax         ;test for wraparound
1418  09A4 C3                           ret
1419                            ;
1420                            getnumber:                      ;get number from input buffer
1421                                                            ;returns:
1422                                                            ;bx = value of number from input
1423                                                            ;al = last character scanned (delimiter)
1424                                                            ;ah = blank flag (0 = blank, 1 = non-blank)
1425  09A5 2BDB                         sub     bx,bx           ;initialize value to 0
1426  09A7 8AE7                         mov     ah,bh           ;initialize blank flag to blank
1427                            getn0:
1428  09A9 E845FA       03F1            call    conin
1429  09AC E885FF       0934            call    delim           ;check for delimiter
1430  09AF 740D         09BE            jz      getnret         ;delimiter found, exit
1431  0981 B104                         mov     cl,4
```

CP/M-86 v.1.1 (vol. III)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
1432
1433  09B3 D3E3                    shl     bx,cl           ;make room for new nibble
1434  09B5 E803FF       0943       call    hexcon          ;convert ascii to binary
1435  09B8 02D8                    add     bl,al           ;add nibble
1436  09BA B401                    mov     ah,1            ;blank flag = non-blank
1437  09BC EBEB         09A9       jmps    getn0
1438                          getnret:
1439  09BE C3                      ret
1440                          ;
1441                          getoffset:                    ;get offset from input line
1442                                                        ;offset is a non-blank number not followed by ':'
1443                                                        ;returns:
1444                                                        ;al = last char scanned (delimiter)
1445                                                        ;bx = value
1446  09BF E8E3FF       09A5       call    getnumber       ;get value to bx
1447  09C2 0AE4                    or      ah,ah           ;check for blank entry
1448  09C4 740D         09D3       jz      geterr          ;don't allow blank entry
1449  09C6 3C3A                    cmp     al,':'          ;check delimiter for ':'
1450  09C8 7409         09D3       jz      geterr          ;don't allow ':' delimiter
1451  09CA C3                      ret
1452                          ;
1453                          getlastoffset:                ;same as getoffset but delimiter must be a cr
1454  09CB E8F1FF       09BF       call    getoffset
1455  09CE 3C0D                    cmp     al,eol
1456  09D0 7501         09D3       jnz     geterr
1457  09D2 C3                      ret
1458                          geterr:
1459  09D3 E909F9       02DF       jmp     err
1460                          ;
1461                          checkword:                    ;check for 'W' following command letter
1462  09D6 E813FA       03F1       call    conin
1463  09D9 3C57                    cmp     al,'W'
1464  09DB 7506         09E3       jnz     chret
1465  09DD C6066D1201              mov     wmode,1
1466  09E2 C3                      ret
1467  09E3 FF0E0412        chret:  dec     conptr          ;to rescan character
1468  09E7 C3                      ret
1469                          ;
1470                          checkreg:                     ;check for valid segment register prefix - <sr>:
1471                                                        ;called with:
1472                                                        ;bl = first char
1473                                                        ;bh = second char
1474                                                        ;returns:
1475                                                        ;si = offset to register, if found
1476                                                        ;zf = found flag (1 = found, 0 = not found)
1477  09E8 80CF80                  or      bh,80h          ;since they are defined like that
1478  09EB BD5D01                  mov     bp,offset segreg        ;point to seg reg names
1479  09EE 2BF6                    sub     si,si           ;initialize index to 0
1480                          check0:
1481  09F0 3B1A                    cmp     bx,[bp+si]      ;is it a seg reg name
1482  09F2 740A         09FE       jz      checkret
1483  09F4 83C602                  add     si,2            ;point to next name
1484  09F7 83FE08                  cmp     si,8            ;check for done (4 seg reg names)
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
SER. #

```
1485
1486  09FA 75F4        09F0        jnz     check0
1487  09FC 08F6                    or      si,si           ;unset zero flag
1488                          checkret:
1489  09FE C3                      ret
1490                          ;
1491                          checksegreg:                ;check for valid seg reg name
1492                                                      ;if found, return contents of seg reg
1493                                                      ;else reset input pointer for rescan
1494                                                      ;returns:
1495                                                      ;dx = seg reg value
1496                                                      ;zf = valid seg reg (1 = valid, 0 = not valid)
1497  09FF FF360412                push    conptr          ;save input pointer for possible rescan
1498  0A03 E8E8F9      03F1        call    conin
1499  0A06 50                      push    ax
1500  0A07 E8E7F9      03F1        call    conin
1501  0A0A 50                      push    ax
1502  0A0B E8E3F9      03F1        call    conin
1503  0A0E 3C3A                    cmp     al,':'          ;valid seg reg must have colon
1504  0A10 5B                      pop     bx
1505  0A11 59                      pop     cx
1506  0A12 8AF9                    mov     bh,cl
1507  0A14 86DF                    xchg    bl,bh
1508  0A16 750C        0A24        jnz     notsr
1509  0A18 E8CDFF      09E8        call    checkreg        ;see if it's a valid name
1510  0A1B 7507        0A24        jnz     notsr
1511  0A1D BDAC11                  mov     bp,offset usercs        ;point to saved user seg reg's
1512  0A20 8B12                    mov     dx,[bp+si]      ;get value of user seg reg
1513  0A22 59                      pop     cx              ;throw away saved input pointer
1514  0A23 C3                      ret
1515                          notsr:
1516  0A24 8F060412                pop     conptr          ;reset for rescan
1517  0A28 C3                      ret
1518                          ;
1519                          getseqandoff:               ;get user location specification
1520                                                      ;may be one of the following forms:
1521                                                      ;<empty>
1522                                                      ;nnnn
1523                                                      ;sr:nnnn
1524                                                      ;aaaa:nnnn
1525                                                      ;if numreq set, <empty> is invalid
1526                                                      ;called with:
1527                                                      ;di = offset of 4 byte area containing <offset><segme
1528                                                      ;numreq must have been initialized by calling routine
1529  0A29 E8D3FF      09FF        call    checksegreg     ;see if there is a segment prefix
1530  0A2C 740D        0A3B        jz      gets0
1531  0A2E E874FF      09A5        call    getnumber       ;no segment prefix, check for number
1532  0A31 0AE4                    or      ah,ah           ;was there one?
1533  0A33 7412        0A47        jz      gets3
1534  0A35 8BD3                    mov     dx,bx           ;move number to dx
1535  0A37 3C3A                    cmp     al,':'          ;was delimiter a ':'
1536  0A39 7509        0A44        jnz     gets2
1537                          gets0:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1538
1539  0A3B E881FF      09BF        call    getoffset       ;segment prefix present, must have number
1540  0A3E 891D                    mov     [di],bx         ;number goes to <offset>
1541  0A40 895502                  mov     2[di],dx        ;first number (or sr) goes to <segment>
1542  0A43 C3                      ret
1543                       gets2:
1544  0A44 8915                    mov     [di],dx         ;only one number, put it in <offset>
1545                       getsret:
1546  0A46 C3                      ret
1547                       gets3:
1548  0A47 F6066C12FF              test    numreq,0ffh     ;blank field, see if ok
1549  0A4C 74F8        0A46        jz      getsret         ;ok, return with no change at [di]
1550  0A4E E98EF8      02DF        jmp     err             ;number was required
1551                       ;
1552                       ;       ******************************
1553                       ;       *                            *
1554                       ;       *     disk i/o routines      *
1555                       ;       *                            *
1556                       ;       ******************************
1557                       ;
1558                       readfile:                       ;read file in fcb into memory described in mcb
1559                                                       ;when done, mcb will have base and length of block to
1560  0A51 A16612                  mov     ax,mcbbase
1561  0A54 A33C12                  mov     startreadseg,ax
1562  0A57 A34012                  mov     endreadseg,ax
1563  0A5A A34412                  mov     dmaseg,ax
1564  0A5D 2BC0                    sub     ax,ax
1565  0A5F A33A12                  mov     startreadoff,ax
1566  0A62 A33E12                  mov     endreadoff,ax
1567                       rf0:
1568  0A65 8B164412                mov     dx,dmaseg
1569  0A69 E823F9      038F        call    setdmab         ;set dma base
1570  0A6C C70642120000            mov     dmaoff,0
1571                       rf1:
1572  0A72 8B164212                mov     dx,dmaoff
1573  0A76 E80CF9      0385        call    setdma          ;set dma offset
1574  0A79 833E681208              cmp     mcblen,8        ;8 paragraphs per sector
1575  0A7E 7239        0AB9        jb      readerr         ;if less than 8 pp's left, not enough memory
1576  0A80 BA5C00                  mov     dx,offset fcb
1577  0A83 E8DCF8      0362        call    readsec
1578  0A86 0AC0                    or      al,al           ;test value returned from bdos
1579  0A88 7528        0AB2        jnz     readdone
1580  0A8A 8306661208              add     mcbbase,8       ;point mcb to next available paragraph
1581  0A8F 832E681208              sub     mcblen,8        ;decrement # of available paragraphs
1582  0A94 A14212                  mov     ax,dmaoff
1583  0A97 047F                    add     al,7fh          ;add sector size - 1
1584  0A99 A33E12                  mov     endreadoff,ax
1585  0A9C A14412                  MOV     AX,DMASEG
1586  0A9F A34012                  MOV     ENDREADSEG,AX
1587  0AA2 810642128000            add     dmaoff,80h      ;increment dma offset
1588  0AA8 75C8        0A72        jnz     rf1             ;if no wrap occurred, simply continue
1589  0AAA 810644120010            add     dmaseg,1000h    ;else increment dma segment
1590  0AB0 EBB3        0A65        jmps    rf0
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                        asmerr:
 1591  0B57 E985F7          02DF          jmp     err
 1592                                 asment:                         ;arrive here on input error in pl/m
 1593  0B5A 8B260712                      mov     sp,asmspsav         ;reset stack to where it was
 1594  0B5E EBE0              0B40        jmps    asm0                ;go back for more input
 1595                                 ;
 1596                                 ;       ******************************
 1597                                 ;       *                            *
 1598  4
 1599  0AC1 D3E8                          shr     ax,cl               ;divide offset by 16 - truncate ls nibble
 1600  0AC3 03064D12                      add     ax,startwriteseg        ;compute absolute paragraph number
 1601  0AC7 A36612                        mov     mcbbase,ax
 1602  0ACA A34412                        mov     dmaseg,ax
 1603  0ACD 8BD8                          mov     bx,ax               ;store start paragraph # in [bx]
 1604  0ACF A14F12                        mov     ax,endwriteoff
 1605  0AD2 D3E8                          shr     ax,cl
 1606  0AD4 03065112                      add     ax,endwriteseg      ;calculate absolute paragraph number of end
 1607  0AD8 2BC3                          sub     ax,bx               ;compute # of paragraphs to write
 1608  0ADA 7250              0B2C        jb      wferr               ;start can't be > end
 1609  0ADC A36812                        mov     mcblen,ax           ;store # paragraphs to write
 1610  0ADF BA5C00                        mov     dx,offset fcb
 1611  0AE2 E878F8            035D        call    delete
 1612  0AE5 F6067112FF                    TEST    ERRMODE,0FFH
 1613  0AEA 7408              0AF4        JZ      WF00
 1614  0AEC FEC0                          INC     AL                  ;DID DELETE RETURN 0FFH?
 1615  0AEE 7504              0AF4        JNZ     WF00                ;IF NOT, OK
 1616  0AF0 0AE4                          OR      AH,AH               ;SEE IF EXTENDED OR PHYSICAL ERROR
 1617  0AF2 7538              0B2C        JNZ     WFERR               ;IF SO, DON'T CONTINUE
 1618                                 WF00:
 1619  0AF4 BA5C00                        mov     dx,offset fcb
 1620  0AF7 E87CF8            0376        call    make
 1621                                 wf0:
 1622  0AFA 8B164412                      mov     dx,dmaseg
 1623  0AFE E88CF8            038F        call    setdmab
 1624  0B01 C706421200000                 mov     dmaoff,0            ;clear dma offset
 1625                                 wf1:
 1626  0B07 8B164212                      mov     dx,dmaoff
 1627  0B0B E877F8            0385        call    setdma
 1628  0B0E BA5C00                        mov     dx,offset fcb
 1629  0B11 E853F8            0367        call    writesec
 1630  0B14 832E681208                    sub     mcblen,8            ;8 paragraphs per sector
 1631  0B19 7210              0B2B        jb      writedone
 1632  0B1B 810642128000                  add     dmaoff,80h          ;increment dma pointer
 1633  0B21 75E4              0B07        jnz     wf1                 ;loop if no wrap occurred
 1634  0B23 810644120010                  add     dmaseg,1000h        ;if wrap occurred, increment dma segment
 1635  0B29 EBCF              0AFA        jmps    wf0
 1636                                 writedone:
 1637  0B2B C3                            ret
 1638                                 wferr:
 1639  0B2C E9B0F7            02DF        jmp     err
 1640                                 ;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1641
1642                            eject
1643                            ;
1644                            ;       ********************************
1645                            ;       *                              *
1646                            ;       *     a - assemble mnemonics   *
1647                            ;       *                              *
1648                            ;       ********************************
1649                            ;
1650                            asm:
1651  0B2F F6060612FF                   test    assempresent,0ffh
1652  0B34 7421         0B57            jz      asmerr
1653  0B36 89260712                     mov     asmspsav,sp     ;save in case of error in pl/m
1654  0B3A BF5812                       mov     di,offset typelloc
1655  0B3D E8E9FE       0A29            call    getsegandoff    ;get start address
1656                            asm0:
1657  0B40 FF365A12                     push    typelseg        ;for pl/m call
1658  0B44 FF365812                     push    typelloc        ;for pl/m call
1659  0B48 E8B517       2300            call    assem           ;returns offset of next available byte
1660  0B4B 3B065812                     cmp     ax,typelloc     ;test for no input
1661  0B4F 7605         0B56            jna     asmret          ;done unless greater than original typelloc
1662  0B51 A35812                       mov     typelloc,ax     ;update typelloc
1663  0B54 EBEA         0B40            jmps    asm0
1664                            asmret:
1665  0B56 C3                           ret
1666                            asmerr:
1667  0B57 E985F7       02DF            jmp     err
1668                            asment:                         ;arrive here on input error in pl/m
1669  0B5A 8B260712                     mov     sp,asmspsav     ;reset stack to where it was
1670  0B5E EBE0         0B40            jmps    asm0            ;go back for more input
1671                            ;
1672                            ;       ******************************
1673                            ;       *                            *
1674                            ;       *    B - BLOCK COMPARE       *
1675                            ;       *                            *
1676                            ;       ******************************
1677                            ;
1678                            BLOCKCOMPARE:
1679  0B60 BF5C12                       mov     di,offset type2loc
1680  0B63 E8C3FE       0A29            call    getsegandoff
1681  0B66 E856FE       09BF            call    getoffset       ;get end offset
1682  0B69 891E6012                     mov     usermax,bx
1683  0B6D 3C0D                         cmp     al,eol
1684  0B6F 7451         0BC2            jz      cmperr          ;need 3 arguments
1685  0B71 2B1E5C12                     sub     bx,type2loc
1686  0B75 724B         0BC2            jb      cmperr          ;error if start > end
1687  0B77 A15E12                       mov     ax,type2seg
1688  0B7A A36412                       mov     userseg2,ax     ;default to same seg as source
1689  0B7D BF6212                       mov     di,offset userloc2
1690  0B80 E8A6FE       0A29            call    getsegandoff    ;get destination address
1691  0B83 3C0D                         cmp     al,eol
1692  0B85 753B         0BC2            jnz     cmperr          ;error if more than 3 arguments
1693                            CMP0:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1694
1695  0BB7 C4365C12              LES     SI,DWORD PTR TYPE2LOC
1696  0BBB 268A04                MOV     AL,ES:[SI]
1697  0BBE C4366212              LES     SI,DWORD PTR USERLOC2
1698  0BB2 263A04                CMP     AL,ES:[SI]
1699  0BB5 7415          0BAC    JZ      CMPCONT
1700  0BB7 E83EF3        0428    CALL    CRLFCHK
1701  0B9A BF5C12                MOV     DI,OFFSET TYPE2LOC
1702  0B9D E82500        0BC5    CALL    PRINTERROR
1703  0BA0 E88CF8        042F    CALL    BLANK
1704  0BA3 E889F8        042F    CALL    BLANK
1705  0BA6 BF6212                MOV     DI,OFFSET USERLOC2
1706  0BA9 E81900        0BC5    CALL    PRINTERROR
1707                       CMPCONT:
1708  0BAC FF065C12              INC     TYPE2LOC
1709  0BB0 A15C12                MOV     AX,TYPE2LOC
1710  0BB3 39066012              CMP     USERMAX,AX
1711  0BB7 7208          0BC1    JC      CMPDONE
1712  0BB9 FF066212              INC     USERLOC2
1713  0BBD 7402          0BC1    JZ      CMPDONE         ;PREVENT WRAPAROUND
1714  0BBF EBC6          0B07    JMPS    CMP0
1715                       ;
1716                       CMPDONE:
1717  0BC1 C3                    RET
1718                       CMPERR:
1719  0BC2 E91AF7        02DF    JMP     ERR
1720                       ;
1721                       PRINTERROR:                     ;PRINT DWORD AT [DI], BYTE POINTED TO BY DWORD
1722  0BC5 C43D                  LES     DI,[DI]
1723  0BC7 268A05                MOV     AL,ES:[DI]
1724  0BCA 50                    PUSH    AX              ;SAVE BYTE AT ES:DI
1725  0BCB E89BF8        0469    CALL    PRINTDWORD
1726  0BCE E85EF8        042F    CALL    BLANK
1727  0BD1 58                    POP     AX
1728  0BD2 E8A7F8        047C    CALL    PRINTBYTE
1729  0BD5 C3                    RET
1730                       ;
1731                       ;       ********************************
1732                       ;       *                              *
1733                       ;       *     d - display memory       *
1734                       ;       *                              *
1735                       ;       ********************************
1736                       ;
1737                       display:
1738  0BD6 C6066C1200            mov     numreq,0        ;ok to have no entries
1739  0BDB A15E12                mov     ax,type2seg
1740  0BDE A30B12                mov     disseg,ax       ;default to type2 seg
1741  0BE1 E8F2FD        09D6    call    checkword
1742  0BE4 F6067012FF            TEST    COL40,0FFH
1743  0BE9 7411          0BFC    JZ      DIS01
1744  0BEB F6066D1201            TEST    WMODE,1
1745  0BF0 7505          0BF7    JNZ     DIS00
1746  0BF2 A01312                MOV     AL,ND40
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1747
1748  0BF5 EB08          0BFF          JMPS    DIS02
1749                          DIS00:
1750  0BF7 A01412                      MOV     AL,NDW40        ;CHARS PER LINE FOR DW IN 40 COL MODE
1751  0BFA EB03          0BFF          JMPS    DIS02
1752                          DIS01:
1753  0BFC A01212                      MOV     AL,ND80         ;16 BYTES PER LINE IN NORMAL MODE
1754                          DIS02:
1755  0BFF A21112                      MOV     LINEMAX,AL
1756  0C02 BF0912                      mov     di,offset disloc
1757  0C05 E821FE        0A29          call    getsegandoff
1758  0C08 3C2C                        cmp     al, ','
1759  0C0A A10B12                      mov     ax,disseg
1760  0C0D A35E12                      mov     type2seg,ax     ;update default type2 seg
1761  0C10 7505          0C17          jnz     dis0            ;must be cr, no dismax entered
1762  0C12 E886FD        09CB          call    getlastoffset   ;get dismax
1763  0C15 EB1A          0C31          jmps    dis1
1764                          dis0:
1765  0C17 8B1E0912                    mov     bx,disloc       ;no dismax entered, calculate default
1766  0C1B A01112                      MOV     AL,LINEMAX
1767  0C1E 8A0E1512                    MOV     CL,NLINES
1768  0C22 F6E1                        MUL     CL
1769  0C24 FEC8                        DEC     AL
1770  0C26 03D8                        ADD     BX,AX
1771  0C28 3B1E0912                    cmp     bx,disloc       ;see if we went over ffff
1772  0C2C 7303          0C31          jnb     dis1
1773  0C2E BBFFFF                      mov     bx,0ffffh       ;set dismax if we wrapped around
1774                          dis1:
1775  0C31 891E0D12                    mov     dismax,bx
1776                          disp3:
1777  0C35 E8F0F7        0428          CALL    CRLFCHK
1778  0C38 C43E0912                    les     di,dword ptr disloc
1779  0C3C 893E0F12                    mov     tdisp,di
1780  0C40 E826F8        0469          call    printdword
1781                          disp4:
1782  0C43 E8E9F7        042F          call    blank
1783  0C46 C4360912                    les     si,dword ptr disloc
1784  0C4A E810F8        045D          call    print8or16
1785  0C4D BE0912                      mov     si,offset disloc
1786  0C50 E846FD        0999          call    inclor2
1787  0C53 7216          0C6B          jb      disp6           ;stop if wrap occurred
1788  0C55 A10912                      mov     ax,disloc
1789  0C58 2B060F12                    sub     ax,tdisp        ;calculate # bytes printed on line
1790  0C5C 3A061112                    CMP     AL,LINEMAX      ;SEE IF LINE FULL
1791  0C60 7409          0C6B          jz      disp6
1792  0C62 A10912                      mov     ax,disloc
1793  0C65 3B060D12                    cmp     ax,dismax       ;check for done
1794  0C69 76D8          0C43          jna     disp4
1795                          disp6:
1796  0C6B E8C1F7        042F          call    blank
1797                          disp7:
1798  0C6E 8E060B12                    mov     es,disseg
1799  0C72 8B360F12                    mov     si,tdisp
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
1800
1801  0C76 268A04              mov   al,es:[si]
1802  0C79 E8D3F7      044F    call  ascout
1803  0C7C FF060F12            inc   tdisp
1804  0C80 7411        0C93    jz    disp8       ;stop if wrap occurred
1805  0C82 A10F12              mov   ax,tdisp
1806  0C85 3B060912            cmp   ax,disloc
1807  0C89 75E3        0C6E    jnz   disp7
1808  0C8B 3B060D12            cmp   ax,dismax
1809  0C8F 7702        0C93    ja    disp8
1810  0C91 EBA2        0C35    jmps  disp3
1811                     disp8:
1812  0C93 C3                  ret
1813                     ;
1814                     ;       ****************************************
1815                     ;       *                                      *
1816                     ;       *     e - load program for execution   *
1817                     ;       *                                      *
1818                     ;       ****************************************
1819                     ;
1820  0C94 434D44              cmd           db    'CMD'
1821                     ;
1822                     execute:
1823  0C97 E857F7      03F1    call  conin
1824  0C9A 3C0D                cmp   al,eol          ;check for no filename
1825  0C9C 7474        0D12    jz    eerr            ;don't allow no filename
1826  0C9E FF0E0412            dec   comptr          ;to rescan character
1827  0CA2 BF5C00              mov   di,offset fcb
1828  0CA5 E8E9F7      0491    call  parse
1829  0CA8 7468        0D12    jz    eerr            ;no '?' or '*' allowed
1830  0CAA 3C0D                cmp   al,eol
1831  0CAC 7564        0D12    jnz   eerr            ;eol must follow filename
1832  0CAE 0E                  push  cs
1833  0CAF 07                  pop   es              ;set es = cs
1834  0CB0 26803E650020        cmp   es:fcb+9, ' '   ;see if filetype blank
1835  0CB6 750B        0CC3    jnz   ex0
1836  0CB8 BE940C              mov   si,offset cmd
1837  0CBB BF6500              mov   di,offset fcb+9
1838  0CBE B90300              mov   cx,3
1839  0CC1 F3A4          rep   movs  al,al           ;set filetype to 'CMD' if empty
1840                     ex0:
1841  0CC3 BA5C00              mov   dx,offset fcb
1842  0CC6 E876F6      033F    call  open            ;see if file exists
1843  0CC9 C6066A12FF          mov   mcbext,0ffh     ;free all allocations below DDT86
1844  0CCE BA6612              mov   dx,offset mcb
1845  0CD1 E8D9F6      03AD    call  freemem         ;free all memory previously allocated under DDT
1846  0CD4 BA5C00              mov   dx,offset fcb
1847  0CD7 E8D8F6      03B2    call  load            ;load user program
1848  0CDA 53                  push  bx              ;save ds base
1849  0CDB E8EFF9      06CD    call  setcpustate
1850  0CDE 5A                  pop   dx              ;get ds base back
1851  0CDF E8ADF6      033F    call  setdmab         ;set dma base
1852  0CE2 BA8000              mov   dx,80h
```

COPYRIGHT 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1853
1854  0CE5 E89DF6      0385          call    setdma          ;default to 80h in user's DS
1855  0CE8 BA5C00                    MOV     DX,OFFSET FCB
1856  0CEB E860F6      034E          CALL    CLOSE
1857  0CEE C6066E1245                mov     mode,'E'
1858  0CF3 BE0000                    mov     si,0
1859  0CF6 BF7212                    mov     di,offset basepagesave
1860  0CF9 1E                        push    ds
1861  0CFA 07                        pop     es              ;set es to ddt's segment
1862  0CFB 2EA1AE11                  mov     ax,user_ds
1863  0CFF 1E                        push    ds              ;save it
1864  0D00 8ED8                      mov     ds,ax
1865  0D02 B93000                    mov     cx,48
1866  0D05 F3A4              rep     movs    al,al           ;copy user's base page into ddt save area
1867  0D07 1F                        pop     ds              ;restore ds
1868  0D08 E8D302      0FDE          call    verify          ;display load info
1869  0D0B FF0E0412                  DEC     CONPTR          ;TO RESCAN CR
1870  0D0F E9CC00      0DDE          JMP     IFCB            ;TO CLEAR FCB
1871                         eerr:
1872  0D12 E9CAF5      02DF          jmp     err
1873                         ;
1874                         ;       ****************************
1875                         ;       *                          *
1876                         ;       *    f - fill memory       *
1877                         ;       *                          *
1878                         ;       ****************************
1879                         ;
1880                         fill:
1881  0D15 E8BEFC      09D6          call    checkword       ;check for 'FW'
1882  0D18 BF5C12                    mov     di,offset type2loc
1883  0D1B E80BFD      0A29          call    getsegandoff
1884  0D1E E89EFC      09BF          call    getoffset       ;get end address
1885  0D21 891E6012                  mov     usermax,bx      ;save end address
1886  0D25 E8A3FC      09CB          call    getlastoffset   ;get fill constant
1887  0D28 F6066D1201                test    wmode,1
1888  0D2D 7504        0D33          jnz     fil0
1889  0D2F 0AFF                      or      bh,bh           ;if not wmode, high byte must be 0
1890  0D31 7525        0D58          jnz     filerr
1891                         fil0:
1892  0D33 8B0E6012                  mov     cx,usermax      ;get end address
1893  0D37 2B0E5C12                  sub     cx,type2loc     ;compare for valid range
1894  0D3B 721B        0D58          jb      filerr          ;error if start > end
1895                         fil1:
1896  0D3D C43E5C12                  les     di,dword ptr type2loc
1897  0D41 8BC3                      mov     ax,bx           ;get fill constant
1898  0D43 E829FC      096F          call    set8or16
1899  0D46 BE5C12                    mov     si,offset type2loc
1900  0D49 E84DFC      0999          call    inclor2
1901  0D4C 7209        0D57          jb      filret          ;stop if wrap occurred
1902  0D4E A15C12                    mov     ax,type2loc
1903  0D51 3B066012                  cmp     ax,usermax
1904  0D55 76E6        0D3D          jbe     fil1
1905                         filret:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1906
1907  0D57 C3                       ret
1908                          filerr:
1909  0D58 E984F5       02DF        jmp     err
1910                          ;
1911                          ;       ************************************
1912                          ;       *                                  *
1913                          ;       *    g - go to user program        *
1914                          ;       *                                  *
1915                          ;       ************************************
1916                          ;
1917                          gouser:
1918  0D5B 2EA1AC11                   mov     ax,usercs
1919  0D5F A31812                     mov     goseg,ax        ;default goseg = usercs
1920  0D62 2EA1B411                   mov     ax,userip
1921  0D66 A31612                     mov     goloc,ax        ;default goloc = userip
1922  0D69 C606C61200                 mov     numreq,0        ;number not required in G command
1923  0D6E BF1612                     mov     di,offset goloc
1924  0D71 E835FC       0A29          call    getsegandoff    ;get start address
1925  0D74 3C0D                       cmp     al,eol
1926  0D76 742B         0DA3          jz      gorestore       ;if eol, no breakpoints set
1927  0D78 A11812                     mov     ax,goseg
1928  0D7B A32912                     mov     brk1seg,ax
1929  0D7E A32B12                     mov     brk2seg,ax      ;defaults for breakpoint segments = goseg
1930  0D81 BF2712                     mov     di,offset brk1loc
1931  0D84 E8A2FC       0A29          call    getsegandoff    ;get first breakpoint
1932  0D87 50                         push    ax              ;save terminating char
1933  0D88 BF2712                     mov     di,offset brk1loc
1934  0D8B E8A5F9       0733          call    setbp           ;save breakpoint in table
1935  0D8E 58                         pop     ax              ;get char
1936  0D8F 3C0D                       cmp     al,eol
1937  0D91 7410         0DA3          jz      gorestore       ;only one breakpoint
1938  0D93 BF2C12                     mov     di,offset brk2loc
1939  0D96 E890FC       0A29          call    getsegandoff    ;get second breakpoint
1940  0D99 3C0D                       cmp     al,eol
1941  0D9B 751C         0DB9          jnz     goerr           ;only 2 breakpoints allowed
1942  0D9D BF2C12                     mov     di,offset brk2loc
1943  0DA0 E890F9       0733          call    setbp           ;set second breakpoint
1944                          gorestore:
1945  0DA3 C606491200                 mov     skipbdos,0      ;make sure it's 0 since we aren't in T/U mode
1946  0DA8 A11812                     mov     ax,goseg
1947  0DAB 2EA3AC11                   mov     usercs,ax       ;usercs = goseg
1948  0DAF A11612                     mov     ax,goloc
1949  0DB2 2EA3B411                   mov     userip,ax       ;userip = goloc
1950  0DB6 E946F3       08FF          jmp     rstore          ;restore user CPU state
1951                          goerr:
1952  0DB9 E838F7       0742          call    bpclear         ;in case any were set
1953  0DBC E920F5       02DF          jmp     err
1954                          ;
1955                          ;
1956                          ;       ************************
1957                          ;       *                      *
1958                          ;       *    h - hex math      *
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1959
1960                        ;       *                       *
1961                        ;       *************************
1962                        ;
1963                        hexmath:
1964  0DBF E8FDFB   09BF            call    getoffset
1965  0DC2 53                       push    bx              ;save first value
1966  0DC3 E805FC   09CB            call    getlastoffset
1967  0DC6 58                       pop     ax              ;get first value
1968  0DC7 50                       push    ax              ;save a copy
1969  0DC8 03C3                     add     ax,bx
1970  0DCA 53                       push    bx              ;save second value
1971  0DCB 50                       push    ax              ;save sum
1972  0DCC E84EF6   041D            call    crlf
1973  0DCF 58                       pop     ax              ;get sum
1974  0DD0 E8A2F6   0475            call    printword       ;print sum
1975  0DD3 E859F6   042F            call    blank
1976  0DD6 5B                       pop     bx              ;get second value
1977  0DD7 58                       pop     ax              ;get first value
1978  0DD8 2BC3                     sub     ax,bx
1979  0DDA E898F6   0475            call    printword       ;print difference
1980  0DDD C3                       ret
1981                        ;
1982                        ;       ****************************************
1983                        ;       *                                      *
1984                        ;       *    i - input file control block      *
1985                        ;       *                                      *
1986                        ;       ****************************************
1987                        ;
1988                        ifcb:
1989  0DDE FF360412                 push    conptr          ;save input pointer
1990  0DE2 BF5C00                   mov     di,offset fcb
1991  0DE5 E8A9F6   0491            call    parse
1992  0DE8 3C0D                     cmp     al,eol
1993  0DEA 7504     0DF0            jnz     i0              ;only one filename
1994  0DEC FF0E0412                 dec     conptr          ;to rescan eol and blank second filename in fcb
1995                        i0:
1996  0DF0 BF6C00                   mov     di,offset fcb2
1997  0DF3 E8A6F6   049C            call    parse2          ;parse second filename
1998  0DF6 1E                       push    ds
1999  0DF7 07                       pop     es              ;point to DDT's ds
2000  0DF8 8F060412                 pop     conptr          ;restore input pointer
2001  0DFC BF8100                   mov     di,81h          ;move command tail to [es]:[di]
2002  0DFF 2BC9                     sub     cx,cx           ;zero count
2003                        i1:
2004  0E01 E8EDF5   03F1            call    conin           ;get char from command tail
2005  0E04 3C0D                     cmp     al,eol          ;end of command tail?
2006  0E06 7404     0E0C            jz      i2
2007  0E08 AA                       stos    al              ;store in user's base page
2008  0E09 41                       inc     cx              ;increment count
2009  0E0A EBF5     0E01            jmps    i1              ;loop until eol
2010                        i2:
2011  0E0C B000                     mov     al,0
```

COPYRIGHT © 1981, 19
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 939
SER. #

```
2012
2013  0E0E AA                          stos    al              ;store 0 at end of string
2014  0E0F 26830E8000                  mov     es:.80h,cl      ;store count at start of buffer
2015  0E14 803E6E1245                  cmp     mode,'E'
2016  0E19 750F          0E2A          jnz     idone           ;if no file loaded with E command, we're done
2017  0E1B BE5C00                      mov     si,offset fcb
2018  0E1E 8BFE                        mov     di,si
2019  0E20 2E8E06AE11                  mov     es,userds
2020  0E25 83C126                      add     cx,38           ;total bytes to move = # in command + 36 (fcb)
2021                                                           ; +2 (0 at end of command and count byte)
2022  0E28 F3A4                rep     movs    al,al          ;move fcb from ddt86 basepage to user's basepage
2023                           idone:
2024  0E2A C3                          ret
2025                           ierr:
2026  0E2B E9B1F4        02DF          jmp     err
2027                           ;
2028                           ;       ********************************
2029                           ;       *                              *
2030                           ;       *    1 - list assembly code    *
2031                           ;       *                              *
2032                           ;       ********************************
2033                           ;
2034                           lasm:
2035  0E2E F6063912FF                  test    disempresent,0ffh
2036  0E33 746C          0EA1          jz      laserr
2037  0E35 C606371200                  mov     lascntsw,0      ;don't use count if end addr specified
2038  0E3A C6066C1200                  mov     numreq,0        ;ok if no entries
2039  0E3F A15A12                      mov     ax,typeiseg
2040  0E42 A33312                      mov     lasseg,ax       ;default to typei seg
2041  0E45 BF3112                      mov     di,offset lasloc
2042  0E48 E8DEFB        0A29          call    getsegandoff
2043  0E4B 3C0D                        cmp     al,eol
2044  0E4D A13312                      mov     ax,lasseg
2045  0E50 A35A12                      mov     typeiseg,ax     ;update default typei seg
2046  0E53 7405          0E5A          jz      las0
2047  0E55 E873FB        09CB          call    getlastoffset   ;if ',', get end address
2048  0E58 EB0E          0E68          jmps    las1
2049                           las0:
2050  0E5A C606371201                  mov     lascntsw,1      ;disassemble fixed # of instructions
2051  0E5F A01512                      MOV     AL,NLINES
2052  0E62 A23B12                      MOV     LASCNT,AL
2053  0E65 BBFFFF                      mov     bx,0ffffh       ;set lasmax to big number
2054                           las1:
2055  0E68 891E3512                    mov     lasmax,bx
2056                           las2:
2057  0E6C 8B3E3112                    mov     di,lasloc
2058  0E70 3B3E3512                    cmp     di,lasmax
2059  0E74 772A          0EA0          ja      lasret
2060  0E76 57                          push    di
2061  0E77 E8AEF5        0428          CALL    CRLFCHK
2062  0E7A 5F                          pop     di
2063  0E7B 8E063312                    mov     es,lasseg
2064  0E7F 06                          push    es
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2065
2066  0EB0 57                          push    di              ;for disem call (PL/M)
2067  0EB1 E8E5F5       0469           call    printdword
2068  0EB4 E8A8F5       042F           call    blank
2069  0EB7 E87605       1400           call    disem
2070  0EBA 3B063112                    cmp     ax,lasloc
2071  0EBE 7210         0EA0           jb      lasret          ;stop if wrap occurred
2072  0EC0 A33112                      mov     lasloc, ax
2073  0EC3 F6063712FF                  test    lascntsw,0ffh
2074  0EC8 74D2         0E6C           jz      las2
2075  0ECA FE0E3812                    dec     lascnt
2076  0ECE 75CC         0E6C           jnz     las2
2077                              lasret:
2078  0ED0 C3                          ret
2079                              laserr:
2080  0ED1 E93BF4       02DF           jmp     err
2081                              ;
2082                              ;       ***************************
2083                              ;       *                         *
2084                              ;       *     m - move block      *
2085                              ;       *                         *
2086                              ;       ***************************
2087                              ;
2088                              move:
2089  0ED4 BF5C12                      mov     di,offset type2loc
2090  0ED7 E87FFB       0A29           call    getsegandoff
2091  0EDA E812FB       09BF           call    getoffset       ;get end offset
2092  0EDD 891E6012                    mov     usermax,bx
2093  0EE1 3C0D                        cmp     al,eol
2094  0EE3 743A         0EEF           jz      moverr          ;need 3 arguments
2095  0EE5 2B1E5C12                    sub     bx,type2loc
2096  0EE9 7234         0EEF           jb      moverr          ;error if start > end
2097  0EEB A15E12                      mov     ax,type2seg
2098  0EEE A36412                      mov     userseg2,ax     ;default to same seg as source
2099  0EF1 BF6212                      mov     di,offset userloc2
2100  0EF4 E862FB       0A29           call    getsegandoff    ;get destination address
2101  0EF7 3C0D                        cmp     al,eol
2102  0EF9 7524         0EEF           jnz     moverr          ;error if more than 3 arguments
2103                              mov0:
2104  0EFB C4365C12                    les     si,dword ptr type2loc
2105  0EFF 268A04                      mov     al,es:[si]      ;get source byte
2106  0F02 C43E6212                    les     di,dword ptr userloc2
2107  0F06 E8A7FA       0980           call    setbyte         ;put destination byte
2108  0F09 FF065C12                    inc     type2loc
2109  0F0D 740F         0F1E           jz      movret          ;don't allow wraparound
2110  0F0F FF066212                    inc     userloc2
2111  0F13 7409         0F1E           jz      movret          ;don't allow wraparound in destination segment
2112  0F15 A15C12                      mov     ax,type2loc
2113  0F18 3B066012                    cmp     ax,usermax      ;check for done
2114  0F1C 76DD         0EFB           jna     mov0
2115                              movret:
2116  0F1E C3                          ret
2117                              moverr:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2118
2119  0EEF E9EDF3       02DF          jmp     err
2120                          ;
2121                          ;       **************************
2122                          ;       *                        *
2123                          ;       *     r - read file      *
2124                          ;       *                        *
2125                          ;       **************************
2126                          ;
2127                          read:
2128  0EF2 E8FCF4       03F1          call    conin           ;get first command char
2129  0EF5 3C0D                       cmp     al,eol          ;check for no input
2130  0EF7 7450         0F49          jz      rerr            ;filename must be included in command
2131  0EF9 FF0E0412                   dec     conptr          ;to rescan first char
2132  0EFD BF5C00                     mov     di,offset fcb
2133  0F00 E88EF5       0491          call    parse
2134  0F03 7444         0F49          jz      rerr            ;no '?' or '*' allowed
2135  0F05 3C0D                       cmp     al,eol
2136  0F07 7540         0F49          jnz     rerr            ;no parameters after filename
2137  0F09 BA5C00                     mov     dx,offset fcb
2138  0F0C E830F4       033F          call    open
2139  0F0F C7063812FFFF               mov     mcblen,0ffffh   ;largest memory request
2140  0F15 BA6612                     mov     dx,offset mcb
2141  0F18 E879F4       0394          call    getmaxmem       ;get size of largest chuck of memory
2142  0F1B BA6612                     mov     dx,offset mcb
2143  0F1E E87DF4       039E          call    allocabsmem     ;allocate block returned from getmaxmem
2144  0F21 E82DFB       0A51          call    readfile        ;read file into memory block
2145  0F24 C6066A1200                 mov     mcbext,0        ;only free memory at mbase
2146  0F29 BA6612                     mov     dx,offset mcb
2147  0F2C E87EF4       03AD          call    freemem         ;free memory not used in read (read updated mcb)
2148  0F2F A13C12                     mov     ax,startreadseg
2149  0F32 A33E12                     mov     type2seg,ax     ;set default type2 segment to file just read
2150  0F35 A33A12                     mov     type1seg,ax     ;also type1 segment
2151  0F38 2BC0                       sub     ax,ax
2152  0F3A A30912                     mov     disloc,ax       ;display pointer offset = 0
2153  0F3D A33112                     mov     lasloc,ax       ;list pointer offset = 0
2154  0F40 C6066E1252                 mov     mode,'R'        ;last disk input was read (not execute)
2155  0F45 E89600       0FDE          call    verify
2156  0F48 C3                         ret
2157                          rerr:
2158  0F49 E993F3       02DF          jmp     err
2159                          ;
2160                          ;       **************************
2161                          ;       *                        *
2162                          ;       *     s - set memory     *
2163                          ;       *                        *
2164                          ;       **************************
2165                          ;
2166                          setmem:
2167  0F4C E887FA       09D6          call    checkword       ;check for 'SW'
2168  0F4F BF5C12                     mov     di,offset type2loc
2169  0F52 E8D4FA       0A29          call    getsegandoff
2170                          set0:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2171
2172  0F55 E8C5F4      041D        call    crlf
2173  0F58 C43E5C12                les     di,dword ptr type2loc
2174  0F5C E80AF5      0469        call    printdword
2175  0F5F E8CDF4      042F        call    blank
2176  0F62 C4365C12                les     si,dword ptr type2loc
2177  0F66 E8F4F4      045D        call    print8or16
2178  0F69 E8C3F4      042F        call    blank
2179  0F6C E863F4      03D7        call    getline
2180  0F6F E87FF4      03F1        call    conin
2181  0F72 3C0D                    cmp     al,eol
2182  0F74 741F        0F95        jz      set2
2183  0F76 3C2E                    cmp     al,'.'
2184  0F78 7423        0F9D        jz      setret
2185  0F7A FF0E0412                dec     conptr          ;to rescan first character
2186  0F7E E84AFA      09CB        call    getlastoffset
2187  0F81 89C3                    mov     ax,bx           ;get new value to ax
2188  0F83 F6066D1201              test    wmode,1
2189  0F88 7504        0F8E        jnz     set1
2190  0F8A 0AFF                    or      bh,bh
2191  0F8C 7510        0F9E        jnz     seterr          ;must be < 256 if not SW
2192                        set1:
2193  0F8E C43E5C12                les     di,dword ptr type2loc
2194  0F92 E8DAF9      096F        call    set8or16
2195                        set2:
2196  0F95 BE5C12                  mov     si,offset type2loc
2197  0F98 E8FEF9      0999        call    inclor2
2198  0F9B 73B8        0F55        jnb     set0
2199                        setret:
2200  0F9D C3                      ret
2201                        seterr:
2202  0F9E E93EF3      02DF        jmp     err
2203                        ;
2204                        ;       ****************************************
2205                        ;       *                                      *
2206                        ;       *    t - trace program execution       *
2207                        ;       *                                      *
2208                        ;       ****************************************
2209                        ;
2210                        trace:
2211  0FA1 C606481201              mov     traceprint,1
2212                        trace0:                         ;untrace enters here with traceprint = 0
2213  0FA6 E848F4      03F1        call    conin
2214  0FA9 3C53                    cmp     al,'S'          ;check for TS
2215  0FAB B401                    mov     ah,1
2216  0FAD 7406        0FB5        jz      tr0             ;if TS, set segflag to 1
2217  0FAF FECC                    dec     ah              ;else set segflag to 0, and
2218  0FB1 FF0E0412                dec     conptr          ;decrement pointer to rescan character
2219                        tr0:
2220  0FB5 88265412                mov     segflag,ah      ;print segment registers or not
2221  0FB9 C70646120100            mov     tracecount,1    ;default to 1 instruction trace
2222  0FBF E8E3F9      09A5        call    getnumber
2223  0FC2 3C0D                    cmp     al,eol
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2224
2225  0FC4 750E        0FD4          jnz     traceerr        ;only 1 parameter allowed
2226  0FC6 0AE4                      or      ah,ah           ;see if a number was entered
2227  0FC8 7404        0FCE          jz      trace1          ;skip if no number typed
2228  0FCA 891E4612                  mov     tracecount,bx   ;store number of instructions to trace
2229                           trace1:
2230  0FCE E8C400      1095          call    xdisp           ;display CPU state
2231  0FD1 E9C8F8      089C          jmp     tracerestore    ;restore user's CPU state and return
2232                           traceerr:
2233  0FD4 E908F3      02DF          jmp     er
2234                           ;
2235                           ;       ****************************************
2236                           ;       *                                      *
2237                           ;       *     u - untraced program execution   *
2238                           ;       *                                      *
2239                           ;       ****************************************
2240                           ;
2241                           untrace:
2242  0FD7 C606481200                mov     traceprint,0
2243  0FDC EBC8        0FA6          jmps    trace0          ;common code with trace command
2244                           ;
2245                           ;       ********************************
2246                           ;       *                              *
2247                           ;       *     v - display file info     *
2248                           ;       *                              *
2249                           ;       ********************************
2250                           ;
2251                           verify:
2252  0FDE A06E12                    mov     al,mode
2253  0FE1 3C52                      cmp     al,'R'
2254  0FE3 7407        0FEC          jz      verifyr
2255  0FE5 3C45                      cmp     al,'E'
2256  0FE7 7421        100A          jz      verifye
2257  0FE9 E9F3F2      02DF          jmp     err     ;neither R nor E command done
2258                           verifyr:
2259  0FEC E82EF4      041D          call    crlf
2260  0FEF BED601                    mov     si,offset readm
2261  0FF2 E848F4      043D          call    printm
2262  0FF5 E825F4      041D          call    crlf
2263  0FF8 C43E3A12                  les     di,dword ptr startreadoff
2264  0FFC E86AF4      0469          call    printdword
2265  0FFF E82DF4      042F          call    blank
2266  1002 C43E3E12                  les     di,dword ptr endreadoff
2267  1006 E860F4      0469          call    printdword
2268  1009 C3                        ret
2269                           ;
2270                           verifye:
2271  100A E810F4      041D          call    crlf
2272  100D B90300                    mov     cx,3
2273  1010 E822F4      0435          call    tabs
2274  1013 BED601                    mov     si,offset readm
2275  1016 E824F4      043D          call    printm          ;print header
2276  1019 B000                      mov     al,0            ;initialize count to 0
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2277
2278                            v0:
2279  101B 50                           push    ax              ;save it
2280  101C 58                           pop     ax              ;get count
2281  101D 50                           push    ax              ;save it
2282  101E E864F6       0685            call    printseginfo    ;print name, start, end of segment if non-zero
2283  1021 58                           pop     ax              ;get count
2284  1022 FEC0                         inc     al              ;increment it
2285  1024 803E771201                   cmp     byte ptr basepagesave+5,1       ;check for 8080 model
2286  1029 7404         102F            jz      verret          ;no more segments if 8080 model
2287  102B 3C08                         cmp     al,8            ;max of 8 segments described in base page
2288  102D 72EC         101B            jb      v0              ;done when count = 8
2289                            verret:
2290  102F C3                           ret
2291                            ;
2292                            ;       ***************************************
2293                            ;       *                                     *
2294                            ;       *     w - write memory block to disk  *
2295                            ;       *                                     *
2296                            ;       ***************************************
2297                            ;
2298                            write:
2299  1030 A13C12                       mov     ax,startreadseg
2300  1033 A34D12                       mov     startwriteseg,ax
2301  1036 A13A12                       mov     ax,startreadoff
2302  1039 A34B12                       mov     startwriteoff,ax
2303  103C A14012                       mov     ax,endreadseg
2304  103F A35112                       mov     endwriteseg,ax
2305  1042 A13E12                       mov     ax,endreadoff
2306  1045 A34F12                       mov     endwriteoff,ax
2307  1048 E8A6F3       03F1            call    conin
2308  104B 3C0D                         cmp     al,eol          ;check for no parameters
2309  104D 7443         1092            jz      werr            ;must have a filename
2310  104F FF0E0412                     dec     comptr          ;to rescan first char
2311  1053 BF5C00                       mov     di,offset fcb
2312  1056 E838F4       0491            call    parse           ;get filename
2313  1059 7437         1092            jz      werr            ;don't allow '?' or '*'
2314  105B 3C0D                         cmp     al,eol
2315  105D 7509         1068            jnz     w0              ;not end of input - must be 2 parameters
2316  105F 803E6E1252                   cmp     mode,'R'        ;see if a file was read in
2317  1064 752C         1092            jnz     werr            ;no file read - must have start, end addresses
2318  1066 EB20         1088            jmps    w1              ;continue with write
2319                            w0:
2320  1068 A15A12                       mov     ax,typeseg
2321  106B A34D12                       mov     startwriteseg,ax        ;set default to userds
2322  106E BF4B12                       mov     di,offset startwriteoff
2323  1071 E8B5F9       0A29            call    getsegandoff    ;get start address
2324  1074 3C0D                         cmp     al,eol
2325  1076 741A         1092            jz      werr            ;need 2 parameters
2326  1078 A14D12                       mov     ax,startwriteseg
2327  107B A35112                       mov     endwriteseg,ax  ;end defaults to start
2328  107E BF4F12                       mov     di,offset endwriteoff
2329  1081 E8A5F9       0A29            call    getsegandoff    ;get end address
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2330
2331  1084 3C0D                      cmp     al,eol
2332  1086 750A          1092        jnz     werr            ;no more than 2 parameters
2333                               w1:
2334  1088 E831FA        0ABC        call    writefile
2335  108B BA5C00                    mov     dx,offset fcb
2336  108E E8B0F2        034E        call    close
2337  1091 C3                        ret
2338                               werr:
2339  1092 E94AF2        02DF        jmp     err
2340                               ;
2341                               ;       ****************************************
2342                               ;       *                                      *
2343                               ;       *     x - display/alter CPU state      *
2344                               ;       *                                      *
2345                               ;       ****************************************
2346                               ;
2347                               xdisp:                          ;display CPU state
2348  1095 E816F7        07AE        CALL    SETDEFSEG           ;SET TYPE1SEG, LASLOC TO CS:IP
2349  1098 F6067012FF                TEST    COL40,0FFH
2350  109D 751A          10B9        JNZ     XD40
2351  109F E89BF5        063D        call    printregheader
2352                               xnohdr:                         ;entry point to display CPU state without header
2353  10A2 F6067012FF                TEST    COL40,0FFH
2354  10A7 7517          10C0        JNZ     XNH40
2355  10A9 E871F3        041D        call    crlf
2356  10AC E885F4        0564        call    printflags
2357  10AF E87DF3        042F        call    blank
2358  10B2 E823F5        05D8        call    printregs
2359  10B5 E8A5F5        065D        call    printinstr      ;disassemble instruction at [cs:ip]
2360  10B8 C3                        ret
2361                               ;
2362                               XD40:
2363  10B9 E86FF5        062B        CALL    PREGHDR1
2364  10BC B001                      MOV     AL,1
2365  10BE EB02          10C2        JMPS    XD0
2366                               XNH40:
2367  10C0 B000                      MOV     AL,0
2368                               XD0:
2369  10C2 50                        PUSH    AX              ;SAVE HEADER/NO HEADER FLAG
2370  10C3 E857F3        041D        CALL    CRLF
2371  10C6 E89BF4        0564        CALL    PRINTFLAGS
2372  10C9 E863F3        042F        CALL    BLANK
2373  10CC E8FAF4        05C9        CALL    PREG1
2374  10CF E843F3        041D        CALL    CRLF
2375  10D2 58                        POP     AX
2376  10D3 FEC8                      DEC     AL
2377  10D5 7506          10DD        JNZ     XD1
2378  10D7 E85BF5        0635        CALL    PREGHDR2
2379  10DA E840F3        041D        CALL    CRLF
2380                               XD1:
2381  10DD E8F3F4        05D3        CALL    PREG2
2382  10E0 E87AF5        065D        CALL    PRINTINSTR
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2383
2384  10E3 C3                          RET
2385                           xcom:
2386  10E4 C606541201                  mov     segflag,1       ;display seg reg's in x command
2387  10E9 E805F3       03F1           call    conin
2388  10EC 3C0D                        cmp     al,eol          ;check for command by itself
2389  10EE 74A5         1095           jz      xdisp           ;if so, simply display CPU state
2390  10F0 A25712                      mov     xtemp,al        ;else save char
2391  10F3 E8F8F2       03F1           call    conin
2392  10F6 3C0D                        cmp     al,eol          ;check for single character after X
2393  10F8 746A         1164           jz      xflag           ;if so, must be a flag name
2394  10FA 8A265712                    mov     ah,xtemp        ;else it's a reg name
2395  10FE 0C80                        or      al,80h          ;since names are declared that way
2396  1100 86C4                        xchg    al,ah           ;since that's how it is in memory
2397  1102 E826F4       052B           call    chkreg          ;check for valid reg name & store number in regnum
2398  1105 E8E9F2       03F1           call    conin
2399  1108 3C0D                        cmp     al,eol
2400  110A 7555         1161           jnz     xerr            ;eol must follow reg name
2401                           x0:
2402  110C E80EF3       041D           call    crlf
2403  110F 880E5512                    mov     cx,regnum
2404  1113 E8EBF4       0601           call    printregname
2405  1116 E816F3       042F           call    blank
2406  1119 8B0E5512                    mov     cx,regnum
2407  111D E8D5F4       05F5           call    printregval
2408  1120 E80CF3       042F           call    blank
2409  1123 E8B1F2       03D7           call    getline
2410  1126 E8C8F2       03F1           call    conin
2411  1129 3C2E                        cmp     al,'.'
2412  112B 746E         119B           jz      xret            ;done when '.' entered
2413  112D FF0E0412                    dec     conptr          ;else rescan character
2414  1131 E871F8       09A5           call    getnumber
2415  1134 3C0D                        cmp     al,eol
2416  1136 7529         1161           jnz     xerr            ;eol must follow number
2417  1138 0AE4                        or      ah,ah           ;see if non-blank entry
2418  113A 7419         1155           jz      xnext           ;if blank, go to next reg
2419  113C 8B0E5512                    mov     cx,regnum
2420  1140 8BC3                        mov     ax,bx           ;get new value
2421  1142 80F908                      cmp     cl,8            ;are we updating cs?
2422  1145 7503         114A           jnz     x1
2423  1147 A35A12                      mov     type1seg,ax     ;if so, update default type1 segment
2424                           x1:
2425  114A 80F909                      cmp     cl,9            ;are we updating ds?
2426  114D 7503         1152           jnz     x2
2427  114F A35E12                      mov     type2seg,ax     ;if so, update default type2 segment
2428                           x2:
2429  1152 E805F4       055A           call    setreg
2430                           xnext:
2431  1155 FF065512                    inc     regnum
2432  1159 833E55120D                  cmp     regnum,totreg
2433  115E 72AC         110C           jb      x0
2434  1160 C3                          ret
2435                           xerr:
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2436
2437  1161 E97BF1      02DF        jmp     err
2438                         ;
2439                         xflag:
2440  1164 A05712                 mov     al,xtemp        ;get flag name
2441  1167 E8D7F3      0541        call    checkflag       ;check for valid flag name
2442  116A E830F2      041D        call    crlf
2443  116D 8B0E5512               mov     cx,regnum       ;restore flag number
2444  1171 E819F4      058D        call    printflagname
2445  1174 E8B8F2      042F        call    blank
2446  1177 8B0E5512               mov     cx,regnum
2447  117B E82BF4      05A9        call    printflagval
2448  117E E8AEF2      042F        call    blank
2449  1181 E853F2      03D7        call    getline
2450  1184 E81EF8      09A5        call    getnumber
2451  1187 3C0D                   cmp     al,eol
2452  1189 75D6        1161        jnz     xerr            ;eol must follow number
2453  118B 0AE4                   or      ah,ah           ;see if non-blank entry
2454  118D 740C        119B        jz      xret            ;if blank, done
2455  118F 83FB01                 cmp     bx,1
2456  1192 77CD        1161        ja      xerr            ;flag value must be 0 or 1
2457  1194 8B0E5512               mov     cx,regnum
2458  1198 E8D0F3      0573        call    setflag
2459                         xret:
2460  119B C3                     ret
2461                         ;
```

CP/M-86 v.1.1 (Vol. III)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
2462
2463                                    eject
2464                            ;       ********************************
2465                            ;       *                              *
2466                            ;       *      d a t a   a r e a       *
2467                            ;       *                              *
2468                            ;       ********************************
2469                            ;
2470                            ;       user regs must be in cseg, others may be in dseg
2471                            ;
2472  119C 0000                 userax  dw      0
2473  119E 0000                 userbx  dw      0
2474  11A0 0000                 usercx  dw      0
2475  11A2 0000                 userdx  dw      0
2476  11A4 0000                 usersp  dw      0
2477  11A6 0000                 userbp  dw      0
2478  11A8 0000                 usersi  dw      0
2479  11AA 0000                 userdi  dw      0
2480  11AC 0000                 usercs  dw      0
2481  11AE 0000                 userds  dw      0
2482  11B0 0000                 userss  dw      0
2483  11B2 0000                 useres  dw      0
2484  11B4 0000                 userip  dw      0
2485  11B6 0000                 userfl  dw      0
2486    119C                    userreg equ     userax
2487                            ;
2488  11B8 0000                 savess  dw      0               ;temp holder for sp
2489  11BA 0000                 savesp  dw      0               ;temp holder for sp
2490                            ;
2491  11BC                      breakfl rs      1               ;break/single step flag (must be in CS)
2492                            ;
2493    11BD                    endcs   equ     $
2494                            ;
2495                                    dseg
2496                                    org     offset endcs
2497                            ;
2498                            ;       CCP STACK LOCATION
2499                            ;
2500  11BD 0000                 CCPSS           DW      0
2501  11BF 0000                 CCPSP           DW      0
2502                            ;
2503                            ;       console buffer declarations
2504                            ;
2505    0040                    conbuffmax      equ     64
2506  11C1 40                   conbuffhdr      db      conbuffmax
2507  11C2 00                   conbuffcnt      db      0
2508  11C3                      conbuff         rs      conbuffmax+1    ;leave room for eol
2509  1204 0000                 conptr          dw      0
2510                            ;
2511                            ;       a command declarations
2512                            ;
2513  1206 01                   assempresent    db      1       ;assembler in memory flag
2514  1207                      asmspsav        rw      1       ;temporary save for stack pointer
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
2515
2516                            ;
2517                            ;       d command declarations
2518                            ;
2519  1209 0000                 disloc          dw      0       ;offset for display
2520  120B 0000                 disseg          dw      0       ;segment for display
2521  120D                      dismax          rw      1       ;end offset of display
2522  120F                      tdisp           rw      1       ;temporary storage for disloc
2523  1211 00                   LINEMAX         DB      0       ;# OF BYTES PER LINE
2524  1212 10                   ND80            DB      16      ;16 BYTES PER LINE IN 80 COL MODE
2525  1213 06                   ND40            DB      6       ;6 BYTES PER LINE IN 40 COL MODE
2526  1214 08                   NDW40           DB      8       ;8 BYTES (4 WORDS) FOR DW IN 40 COL MODE
2527  1215 0C                   NLINES          DB      12      ;DEFAULT NUMBER OF LINES FOR L, D COMMANDS
2528                            ;
2529                            ;       g command declarations
2530                            ;
2531  1216 0000                 goloc           dw      0
2532  1218 0000                 goseg           dw      0
2533  121A                      vectorsave      rs      12      ;save area for bytes at 0004h to 000fh
2534  1226 00                   bpcnt           db      0       ;breakpoint count
2535  1227 0000                 brk1loc         dw      0
2536  1229 0000                 brk1seg         dw      0
2537  122B 00                   brk1byt         db      0
2538  122C 0000                 brk2loc         dw      0
2539  122E 0000                 brk2seg         dw      0
2540  1230 00                   brk2byt         db      0
2541                            ;
2542                            ;       l command declarations
2543                            ;
2544  1231 0000                 lasloc          dw      0
2545  1233 0000                 lasseg          dw      0
2546  1235 0000                 lasmax          dw      0
2547  1237                      lascntsw        rb      1       ;# instructions specified or not
2548  1238                      lascnt          rb      1       ;number of instructions to disassemble
2549  1239 01                   disempresent    db      1       ;disassembler in memory flag
2550                            ;
2551                            ;       r command declarations
2552                            ;
2553  123A                      startreadoff    rw      1       ;offset where file read starts
2554  123C                      startreadseg    rw      1       ;segment where file read starts
2555  123E                      endreadoff      rw      1       ;offset where file read ends
2556  1240                      endreadseg      rw      1       ;segment where file read ends
2557  1242                      dmaoff          rw      1       ;offset of 20-bit dma address
2558  1244                      dmaseg          rw      1       ;segment of 20-bit dma address
2559                            ;
2560                            ;       t/u command declarations
2561                            ;
2562  1246                      tracecount      rw      1       ;number of instructions to trace
2563  1248                      traceprint      rb      1       ;display CPU state on each step flag
2564  1249 00                   skipbdos        db      0       ;set when trace suspended during BDOS call
2565  124A 00                   USERIFOFF       DB      0       ;SET WHEN DDT86 MUST REENABLE USER IF
2566                            ;
2567                            ;       w command declarations
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
2568
2569                            ;
2570  124B                      startwriteoff   rw      1       ;offset where file write starts
2571  124D                      startwriteseg   rw      1       ;segment where file write starts
2572  124F                      endwriteoff     rw      1       ;offset where file write ends
2573  1251                      endwriteseg     rw      1       ;segment where file write ends
2574                            ;
2575                            ;       x command declarations
2576                            ;
2577  1253 00                   NREG            DB      0       ;CURRENT NUMBER OF REGISTER NAMES TO DISPLAY
2578                                                            ;(MAY DIFFER FOR 40 COLUMN MODE)
2579    000D                    TOTREG          EQU     13      ;TOTAL NUMBER OF REGISTER NAMES
2580    0009                    nflag           equ     9       ;number of flag names
2581  1254 01                   segflag         db      1       ;print segment register flag
2582  1255                      regnum          rw      1       ;temp for reg/flag number
2583  1257                      xtemp           rb      1       ;temp for first char of reg name
2584                            ;
2585  1258 0000                 typeiloc        dw      0       ;offset for type 1 commands
2586  125A 0000                 typeiseg        dw      0       ;segment for type 1 commands
2587  125C 0000                 type2loc        dw      0       ;offset for type 2 commands
2588  125E 0000                 type2seg        dw      0       ;segment for type 2 commands
2589  1260                      usermax         rw      1
2590  1262                      userloc2        rw      1
2591  1264                      userseg2        rw      1
2592                            ;
2593                            ;       memory control block declarations
2594                            ;
2595  1266                      mcb             rs      0       ;used in reading/writing file
2596  1266 0000                 mcbbase         dw      0       ;segment of memory block
2597  1268 0000                 mcblen          dw      0       ;length of memory block
2598  126A 00                   mcbext          db      0       ;returned value from bdos memory functions
2599                            ;
2600  126B                      sysif           rb      1       ;system interrupt flag
2601  126C                      numreq          rb      1       ;number required or optional
2602  126D                      wmode           rb      1       ;set for DW, FW and SW commands
2603  126E 49                   mode            db      'I'     ;last disk access with 'R' or 'E' command
2604  126F 00                   savevecflag     db      0       ;save/restore bp and ss vectors, or not
2605  1270 00                   COL40           DB      0       ;PATCHED TO 1 FOR 40 COLUMN CONSOLE
2606  1271 00                   ERRMODE         DB      0       ;SET IF BDOS RETURN ERR MODE SET (V3.0)
2607                            ;
2608  1272                      basepagesave    rb      48      ;copy of user's base page
2609                            ;
2610                            ;       parsing declarations
2611                            ;
2612  12A2 203D2E2C3A3B         delims          db      ' ', '=', '.', ',', ':', ';', '[', ']', '<', '>', eol
2613       5B5D3C3E0D
2614    000B                    ndelims         equ     offset $ - offset delims
2615  12AD 00                   lastchar        db      0
2616  12AE 0000                 fcbadr          dw      0       ;temp storage for fcb address
2617                            ;
2618  12B0                                      rs      stsize  ;stack size
2619  1310                      stackp          rs      0       ;top of stack
2620                            ;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

```
2621
2622                                    if      debug
2623                            ;
2624                            biosentryoff    dw      0a00h   ;empty part of ccp (hopefully)
2625                            biosentryseg    dw      0       ;same as bdos segment
2626                            ;
2627                                    endif
2628                            ;
2629                                    end
2630
2631
2632  END OF ASSEMBLY.  NUMBER OF ERRORS:   0.  USE FACTOR: 32%
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
ALLOCABSMEM     039E L   483# 2143
ASCEND          0459 L   664  667#
ASCOUT          044F L   660# 1802
ASM0            0B40 L  1594 1656# 1663  1670
ASMENT          0B5A L    45 1592# 1668#
ASMERR          0857 L  1652 1666#
ASMORG          2300 N    22#   35
ASMRET          0B56 L  1661 1664#
ASMSPSAV        1207 V  1593 1653  1669  2514#
ASSEM           2300 L    36# 1659
ASSEMPRESENT    1206 V  1651 2513#
ASSM            0B2F L   298 1650#
BADVER          0989 L  1397 1401#
BASEPAGESAVE    1272 V  1027 1859  2285  2608#
BDOS            031D L   332#  334   363   383   390   405   412   416   427   437
                         441   445   455   465   469   473   477   485   499   503
                         577
BDOSI           00E0 N    48    53#  219   336
BDOSINT         0112 L    48#   50    51
BDOSINTLOC      0112 N    50# 1270
BDOSINTNUM      0113 N    51#  146
BLANK           042F L   570   633#  641   934   982  1047  1052  1703  1704  1726
                        1782  1796  1975  2068  2175  2178  2265  2357  2372  2405
                        2408  2445  2448
BLOCKCOMPARE    0B60 L   299 1678#
BPCLEAR         0742 L  1117# 1241  1952
BPCLOOP         074B L  1121# 1130
BPCNT           1226 V  1113  1119  1131  2534#
BPEND           075C L  1122  1132#
BPV             075D L  1134# 1302
BPVECT          0764 L   212  1137#
BPVECTRET       0780 L  1136  1151#
BREAK0          0855 L  1239#
BREAKADDR       07C1 L  1188# 1246
BREAKENTRY      07D7 L  1147  1178#
BREAKFL         11BC V  1199  1202  1236  2491#
BRK1BYT         122B V  2537#
BRK1LOC         1227 V  1120  1276  1277  1930  1933  2535#
BRK1SEG         1229 V  1274  1928  2536#
BRK2BYT         1230 V  2540#
BRK2LOC         122C V  1938  1942  2538#
BRK2SEG         122E V  1929  2539#
BUFF            0080 V    29#  166
CCPSP           11BF V   189  1068  2501#
CCPSS           11BD V   188  1066  2500#
CCTRET          024C L   170   183#
CHECK0          09F0 L  1480# 1486
CHECKCMDTAIL    022D L   165#  254
CHECKERR        0557 L   827   839   845#
CHECKFLAG       0541 L   833# 2441
CHECKREG        09E8 L  1470# 1509
CHECKRET        09FE L  1482  1488#
CHECKSEGREG     09FF L  1491# 1529
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
CHECKWORD       09D6 L  1461# 1741  1881  2167
CHKREG          052B L   819# 2397
CHRET           09E3 L  1464  1467#
CLOSE           034E L   424# 1856  2336
CLOSEERR        0358 L   429   431#
CLOSEM          0189 V   114#  432
CMD             0C94 V  1820# 1836
CMPO            0B87 L  1693# 1714
CMPCONT         08AC L  1699  1707#
CMPDONE         0BC1 L  1711  1713  1716#
CMPERR          0BC2 L  1684  1686  1692  1718#
COL40           1270 V  1742  2349  2353  2605#
CONBUFF         11C3 V   176   555   561   589   597  2508#
CONBUFFCNT      11C2 V   587  2507#
CONBUFFIDR      11C1 V   585  2506#
CONBUFFMAX      0040 N   171   173   553  2505# 2506  2508
CONIN           03F1 L    42   269   596#  749   755   776  1428  1462  1498  1500
                       1502  1823  2004  2128  2180  2213  2307  2387  2391  2398
                       2410
CONOUT          0326 L   267   289   353   370#  380#  518   539   572   623   625
                        635   652   657   668   685   708   883   903   915  1191
CONPTR          1204 V   542   553   556   558   562   564   567   569   591   598
                        600   764   766  1467  1497  1516  1826  1869  1989  1994
                       2000  2131  2185  2218  2310  2413  2509#
CONSIN          0320 L   349#  361#  544   613   617
CONSTAT         0334 L   395#  403#  610
COPYRIGHT       0114 V    72#
CR              000D N    62#   63   114   117   121   123   127   132   135   551
                        622
CRLF            041D L   264   287   540   621#  629   971  1013  1044  1189  1972
                       2172  2259  2262  2271  2355  2370  2374  2379  2402  2442
CRLFCHK         0428 L   628# 1700  1777  2061
CTABLE          02E9 V   283   278#
CTLCHEK         0408 L   609#  630  1262
CTLEXIT         041C L   612   618#
CTLS            0013 N    64#  614
D0              027D L   204   206#
D1              028A L   211   214#
D2              02A0 L   247   251#
DATE            0143 V    83#
DDT86           024D L    41   186#
DDTBDOSINTNUM   031E N   148   334#
DDTORG          0100 N    20#    21    22    39
DEBUG           0000 N    18#   215   339   347   359   368   378   393   401   535
                        581  2622
DELETE          035D L   435# 1611
DELIM           0934 L  1331# 1429
DELIMS          12A2 V   770  2612# 2614
DELRET          0942 L  1333  1335  1337  1339#
DIS0            0C17 L  1761  1764#
DIS00           0BF7 L  1745  1749#
DIS01           08FC L  1743  1752#
DIS02           0BFF L  1748  1751  1754#
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
DIS1            0C31 L  1763  1772  1774#
DISEN           1400 L    32#  1015  2069
DISEMPRESENT    1239 V  1006  2035  2549#
DISLOC          1209 V  1756  1765  1771  1778  1783  1785  1788  1792  1806  2152
                        2519#
DISMAX          120D V  1775  1793  1808  2521#
DISP3           0C35 L  1776#  1810
DISP4           0C43 L  1781#  1794
DISP6           0C6B L  1787  1791  1795#
DISP7           0C6E L  1797#  1807
DISP8           0C93 L  1804  1809  1811#
DISPLAY         0BD6 L   301  1737#
DISSEG          120B V  1740  1759  1798  2520#
DMAOFF          1242 V  1570  1572  1582  1587  1624  1626  1632  2557#
DMASEG          1244 V  1563  1568  1585  1589  1602  1622  1634  2558#
EERR            0D12 L  1825  1829  1831  1871#
ENDCS           11BD L  2493#  2496
ENDREADOFF      123E V  1566  1584  2266  2305  2555#
ENDREADSEG      1240 V  1562  1586  2303  2556#
ENDWRITEOFF     124F V  1604  2306  2328  2572#
ENDWRITESEG     1251 V  1606  2304  2327  2573#
EOL             000D N    63#  180   270   563   590   753  1332  1455  1683  1691
                        1824  1830  1925  1936  1940  1992  2005  2043  2093  2101
                        2129  2135  2181  2223  2308  2314  2324  2331  2388  2392
                        2399  2415  2451  2612
ERR             02DF L   275   277   286#  300   307   308   311   312   313   314
                         323   324   746   846  1354  1459  1550  1591  1639  1667
                        1719  1872  1909  1953  2026  2080  2119  2158  2202  2233
                        2257  2339  2437
ERRM            038E L   422   433   451   461   495   510#
ERRMODE         1271 V   252  1612  2606#
EXO             0CC3 L  1835  1840#
EXECUTE         0C97 L   182   302  1822#
FCB             005C V    27#  1576  1610  1619  1628  1827  1834  1837  1841  1846
                        1855  1990  2017  2132  2137  2311  2335
FCB2            006C V    28#  1996
FCBADR          12AE V   728   806  2616#
FIL0            0D33 L  1888  1891#
FIL1            0D3D L  1895#  1904
FILERR          0D58 L  1890  1894  1908#
FILL            0D15 L   303  1880#
FILLBL          0517 L   736   788   797#
FILLDONE        051D L   800   802#
FILLFIELD       04F5 L   732   739   775#  781   792
FILL0           0513 L   780   793#
FILLTYPE        04B9 L   735   738#
FILLX           0519 L   795   799#
FILRET          0D57 L  1901  1905#
FLAGBITS        0170 V   101#  868   889
FLAGNAME        0167 V    99#  835   880   907
FREEMEM         03AD L   497#  1845  2147
GETERR          09D3 L  1448  1450  1456  1458#
GETFLAG         0597 L   886#  897   910
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
GETLASTOFFSET    09CB L  1453#  1762  1886  1966  2047  2186
GETLINE          03D7 L   268   524   541#  583#  2179  2409  2449
GETMAXMEM        0394 L   475#  2141
GETNO            09A9 L  1427#  1437
GETNRET          09BE L  1430  1438#
GETNUMBER        09A5 L  1420#  1446  1531  2222  2414  2450
GETOFFSET        09BF L  1441#  1454  1539  1681  1884  1964  2091
GETS0            0A3B L  1530  1537#
GETS2            0A44 L  1536  1543#
GETS3            0A47 L  1533  1547#
GETSEGANDOFF     0A29 L  1519#  1655  1680  1690  1757  1883  1924  1931  1939  2042
                         2090  2100  2169  2323  2329
GETSRET          0A46 L  1545#  1549
GOERR            0DB9 L  1941  1951#
GOLOC            1216 V  1921  1923  1948  2531#
GORESTORE        0DA3 L  1926  1937  1944#
GOSEG            1218 V  1919  1927  1946  2532#
GOUSER           0D5B L   304  1917#
HEXCON           0943 L  1342#  1434
HEXERR           0954 L  1348  1350  1353#
HEXMATH          0DBF L   305  1963#
HEXRET           0953 L  1345  1351#
I0               0DF0 L  1993  1995#
I1               0E01 L  2003#  2009
I2               0E0C L  2006  2010#
IDONE            0E2A L  2016  2023#
IERR             0E2B L  2025#
IFCB             0DDE L   306  1870  1983#
IFMASK16         0200 N    58#   195
IFMASK3          0002 N    59#   203  1229
INC10R2          0999 L  1411#  1786  1900  2197
KWAIT            0419 L   615   617#
LAS0             0E5A L  2046  2049#
LAS1             0E68 L  2048  2054#
LAS2             0E6C L  2056#  2074  2076
LASCNT           1238 V  2052  2075  2548#
LASCNTSW         1237 V  2037  2050  2073  2547#
LASERR           0EA1 L  2036  2079#
LASLOC           1231 V  1099  1183  2041  2057  2070  2072  2153  2544#
LASMAX           1235 V  2055  2058  2546#
LASMORG          1400 N    21#    31
LASRET           0EA0 L  2059  2071  2077#
LASSEG           1233 V  2040  2044  2063  2545#
LASSM            0E2E L   309  2034#
LASTCHAR         12AD V   734   742   750   758   777  2615#
LF               000A N    61#   114   117   121   123   127   132   135   624
LINEMAX          1211 V  1755  1766  1790  2523#
LOAD             03B2 L   501#  1847
LOADERR          02BB L   505   507#
LOADM            0197 V   117#  508
MAKE             0376 L   453#  1620
MAKEERR          0380 L   457   459#
MAKEM            01AC V   121#  460
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
MCB             1266 V  1844  2140  2142  2146  2595#
MCBBASE         1266 V  1560  1580  1601  2596#
MCBEXT          126A V  1843  2145  2598#
MCBLEN          1268 V  1574  1581  1609  1630  2139  2597#
MEMERR          03A8 L   480   493#
MEMM            01B6 V   123#  494
MODE            126E V  1857  2015  2154  2252  2316  2603#
MOVQ            0ECB L  2103#  2114
MOVCOM          023D L   172   174#
MOVE            0EA4 L   310  2088#
MOVERR          0EEF L  2094  2096  2102  2117#
MOVRET          0EEE L  2109  2111  2115#
NBPS            0002 N    56#
ND40            1213 V  1746  2525#
ND80            1212 V  1753  2524#
NDELIMS         000B N   771  2614#
NDW40           1214 V  1750  2526#
NFLAG           0009 N   834   840  2530#
NLINES          1215 V  1767  2051  2527#
NOT8080         05FE L  1078  1081#
NOTAST          0504 L   779   782#
NOTSR           0A24 L  1508  1510  1515#
NREG            1253 V   920   928   938   979   988   999  2577#
NUMREQ          126C V   231  1548  1738  1922  2038  2601#
OPEN            033F L   414#  1842  2138
OPENERR         0349 L   418   420#
OPENM           01CD V   127#  421
PARSE           0491 L   717#  1828  1991  2133  2312
PARSE2          049C L   726#  1997
PARSEERR        04C3 L   745#  739
PARSERET        04BC L   737   740#
PDELIM          04EC L   769#  735
PEROUT          0457 L   662   665#
PF0             0567 L   858#  864
PF2             05B2 L   899   901#
P10             067A L  1012  1014#
P11             067E L  1007  1017#
PLMCONOUT       03C4 L    43   514#
PLMGETLINE      03D1 L    44   522#
PLMSET          0957 L    46  1356#
PNAME           05C5 L   912   914#
PQUIT           0449 L   650   655#
PR0             05E0 L   921   929#  939
PR00            05DB L   924   927#
PR2             05ED L   931   936#
PREG1           05C9 L   918#  2373
PREG2           05D3 L   922#  2381
PREGHDR1        062B L   977#  2363
PREGHDR2        0635 L   981#  2378
PRH0            0645 L   780   989#  1000
PRH00           0640 L   984   987#
PRH1            0655 L   991   997#
PRINT8OR16      045D L   671#  1784  2177
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
PRINTBYTE        047C L   674   692   695# 1019  1728
PRINTDWORD       0469 L   677# 1051  1060  1195  1408  1725  1780  2067  2174  2264
                          2267
PRINTERROR       0BC5 L  1702  1706  1721#
PRINTFLAG        05B6 L   860   906#
PRINTFLAGNAME    058D L   879# 2444
PRINTFLAGS       0564 L   856# 2356  2371
PRINTFLAGVAL     05A9 L   896# 2447
PRINTINSTR       065D L  1003# 2359  2382
PRINTIT          061C L   961   963   966#
PRINTM           043D L   242   511   646#  654   954  1046  1405  2261  2275
PRINTNIBBLE      0485 L   699   702#
PRINTREGHEADER   063D L   985# 2351
PRINTREGNAME     0601 L   950#  993  2404
PRINTREGS        05D8 L   925# 2358
PRINTREGVAL      05F5 L   933   942# 2407
PRINTSEGINFO     0685 L  1022# 2282
PRINTWORD        0475 L   675   683   688#  947  1974  1979
PSIRET           06CC L  1035  1062#
PSRET            096C L  1363  1365#
RDCONBUFF        032E L   388#  586
READ             0EF2 L   315  2127#
READDONE         0AB2 L  1579
READERR          0AB9 L  1575
READFILE         0A51 L  1558# 2144
READM            01D6 V   129# 2260  2274
READSEC          0362 L   439# 1577
REGNAME          014D V    85#  823   951
REGNUM           1255 V   830   842  2403  2406  2419  2431  2432  2443  2446  2457
                          2582#
RERR             0F49 L  2130  2134  2136  2157#
RESTOREVECT      0796 L  1169# 1235
RF0              0A65 L  1567# 1590
RF1              0A72 L  1571# 1588
RSTORE           08FF L  1280  1301# 1950
RVRET            07AD L  1171  1173#
S0               04E3 L   757   763#
S1               04E7 L   754   765#
SAV0             0843 L  1230  1232#
SAVECPU          07E5 L  1200  1203#
SAVESP           11BA V  1309  1318  2489#
SAVESS           11B8 V  1315  1317  2488#
SAVEVECFLAG      126F V   210  1135  1170  2604#
SAVEVECT         0781 L  1138  1154#
SC1              0715 L  1087  1089#
SCANQ            051E L   741   805#
SEGFLAG          1254 V   960  1011  2220  2386  2581#
SEGNAMES         0179 V   104# 1042
SEGREG           015D V    93# 1478
SET0             0F55 L  2170# 2178
SET1             0F8E L  2189  2192#
SET2             0F95 L  2182  2195#
SET80R16         096F L  1368# 1898  2194
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
SETBDOSINT      020A L   145#  208
SETBP           0733 L  1108#  1278  1934  1943
SETBYTE         0980 L  1361   1370  1387  1394#  2107
SETCPUSTATE     06CD L  1065#  1849
SETDEFSEG       07AE L  1181#  1194  2348
SETDMA          0385 L   463#  1573  1627  1854
SETDMAB         038F L   471#  1569  1623  1851
SETDONE         072B L  1080   1092  1097#
SETERR          0F9E L  2191   2201#
SETERRMODE      038A L   249    467#
SETFLAG         0573 L   867#  2458
SETMEM          0F4C L   316   2166#
SETREG          055A L   848#  2429
SETRET          0F9D L  2184   2199#
SETUPDISK       04C6 L   730    748#   752
SETUPHDR        061E L   970#   978    986
SETWORD         0976 L  1373#
SIGNON          013A V    80#   241
SKIPBDOS        1249 V  1242   1250  1279  1945  2564#
SRET            0988 L  1391   1398#
SSENTRY         07DF L  1142   1201#
SST0            086F L  1237   1251#
SST00           086A L  1243   1249#
SST1            0889 L  1255   1259#
STACKP          1310 V   202    263  1227  2619#
START           02A7 L   262#   271   273   284   290   512   616  1247  1364  1409
STARTREADOFF    123A V  1565   2263  2301  2553#
STARTREADSEG    123C V  1561   2148  2299  2554#
STARTWRITEOFF   124B V  2302   2322  2570#
STARTWRITESEG   124D V  1600   2300  2321  2326  2571#
STSIZE          0060 N    55#  2618
SVRET           0795 L  1165#
SYSIF           126B V   196    203  1229  2600#
TABS            0455 L   639#   643   973   995  2273
TDISP           120F V  1779   1789  1799  1803  1805  2522#
TESTREGCL       060C L   930    958#  990
TOTREG          000D N   822    828  2432  2579#
TR0             0FB5 L  2216   2219#
TR00            08C6 L  1272   1281#
TR01            08FB L  1285   1291  1293  1295  1297  1299#
TRACE           0FA1 L   317   2210#
TRACE0          0FA6 L  2212#  2243
TRACE1          0FCE L  2227   2229#
TRACECOUNT      1246 V  1260   2221  2228  2562#
TRACEDONE       0864 L  1245#  1261
TRACEERR        0FD4 L  2225   2232#
TRACEPRINT      1248 V  1263   2211  2242  2563#
TRACERESTORE    089C L  1264   1267#  2231
TYPE1LOC        1258 V  1654   1658  1660  1662  2585#
TYPE1SEG        125A V  1074   1084  1185  1359  1657  2039  2045  2150  2320  2423
                        2586#
TYPE2LOC        125C V  1679   1685  1695  1701  1708  1709  1882  1893  1896  1899
                        1902   2089  2095  2104  2108  2112  2168  2173  2176  2193
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                    2196  2587#
TYPE2SEG        125E V  1075  1193  1687  1739  1760  2097  2149  2427  2588#
UNTRACE         0FD7 L   318  2241#
UPPER           03FD L   602#
UPRET           0407 L   603   605   607#
USERAX          119C V  1204  2472# 2486
USERBP          11A6 V  1210  2477#
USERBX          119E V  1205  2473#
USERCS          11AC V  1004  1072  1083  1134  1219  1268  1287  1320  1511  1918
                    1947  2480#
USERCX          11A0 V  1206  2474#
USERDI          11AA V  1209  2479#
USERDS          11AE V  1071  1192  1213  1862  2019  2431#
USERDX          11A2 V  1207  2475#
USERES          11B2 V  1073  1088  1212  2483#
USERFL          11B6 V   197   875   876   893  1222  1252  1257  1282  1300  1319
                    2435#
USERIFOFF       124A V  1254  1256  1298  2565#
USERIP          11B4 V  1005  1098  1182  1217  1240  1269  1288  1321  1920  1949
                    2484#
USERLOC2        1262 V  1689  1677  1705  1712  2099  2106  2110  2590#
USERMAX         1260 V  1682  1710  1885  1892  1903  2092  2113  2589#
USERREG         119C V   850   943  1304  2486#
USERSEG2        1264 V  1688  2098  2591#
USERSI          11A8 V  1208  2478#
USERSP          11A4 V   191  1069  1095  1211  1233  2476#
USERSS          11B0 V   190  1067  1093  1214  2482#
V0              101B L  2278# 2288
VECTORSAVE      121A V  1156  1172  2533#
VERIFY          0FDE L   320  1868  2155  2251#
VERIFYE         100A L  2256  2270#
VERIFYR         0FEC L  2254  2258#
VERM            01E6 V   132# 1404
VERRET          102F L  2286  2289#
VERSION         033A L   244   410#
W0              1068 L  2315  2319#
W1              1088 L  2318  2333#
WERR            1092 L  2309  2313  2317  2325  2332  2338#
WF0             0AFA L  1621# 1635
WF00            0AF4 L  1613  1615  1618#
WF1             0B07 L  1625# 1633
WFERR           0B2C L  1608  1617  1638#
WMODE           126D V   282   673  1369  1413  1465  1744  1887  2188  2602#
WRITE           1030 L   321  2298#
WRITEDONE       0B2B L  1631  1636#
WRITEERR        0371 L   447   449#
WRITEFILE       0ABC L  2334
WRITEM          01FB V   135# 450
WRITESEC        0367 L   443# 1629
X0              110C L  2401# 2433
X1              114A L  2422  2424#
X2              1152 L  2426  2428#
XCOM            10E4 L   322  2335#
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

XD0	10C2	L	2365	2368#			
XD1	10DD	L	2377	2380#			
XD40	1089	L	2350	2362#			
XD1SP	1095	L	2230	2347#	2389		
XERR	1161	L	2400	2416	2435#	2452	2456
XFLAG	1164	L	2393	2439#			
XNEXT	1155	L	2418	2430#			
XNH40	10C0	L	2354	2366#			
XNOHDR	10A2	L	1265	2352#			
XRET	119B	L	2412	2454	2459#		
XTEMP	1257	V	2390	2394	2440	2583#	

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____